

# Einbeziehung von Elementen der 3D-Computergrafik in den Mathematikunterricht der Sekundarstufe II im Stoffgebiet Analytische Geometrie

## HABILITATIONSSCHRIFT

zur Erlangung der Lehrbefähigung  
für das Fach Didaktik der Mathematik

vorgelegt dem Rat der  
Mathematisch-Naturwissenschaftlichen Fakultät II  
Humboldt-Universität zu Berlin

von

Dr. rer. nat. Andreas Filler  
geboren am 09.09.1963 in Roßwein

Präsident der Humboldt-Univer-  
sität zu Berlin:  
Prof. Dr. Christoph Marksches

Dekan der Mathematisch-Natur-  
wissenschaftlichen Fakultät II:  
Prof. Dr. Wolfgang Coy

Gutachter:

1. Prof. Dr. Jürg Kramer
2. Priv.-Doz. Dr. Ingmar Lehmann
3. Prof. Dr. Konrad Polthier
4. Prof. Dr. Klaus Volkert

eingereicht am: 13. April 2006

Tag des öffentlichen Vortrags: 27. Februar 2007



# Vorwort

*Die dreidimensionale Computergrafik übt auf viele Jugendliche einen großen Reiz aus und weist zudem eine Reihe interessanter Bezüge zu den Inhalten des Mathematikunterrichts in der Sekundarstufe II auf. Auf der Grundlage einer Diskussion von Zielen und Defiziten des Unterrichts im Stoffgebiet Analytische Geometrie sowie einer zusammenfassenden Darstellung mathematischer Grundlagen der 3D-Computergrafik werden Vorschläge für deren Einbeziehung in den Unterricht unterbreitet.*

Die zentrale Zielstellung der vorliegenden Arbeit besteht darin, Wege aufzuzeigen, durch die Einbeziehung von Elementen der 3D-Computergrafik den Unterricht in analytischer Geometrie praxisnäher, anschaulicher und attraktiver zu gestalten. Dafür sprechen vor allem folgende Gründe:

- Dreidimensionale computergrafische Darstellungen treten mittlerweile in den verschiedensten Lebensbereichen auf und sind für viele Schülerinnen und Schüler<sup>1</sup> sehr reizvoll.
- Die gegenwärtigen Curricula und der auf ihrer Grundlage durchgeführte Unterricht in analytischer Geometrie weisen gravierende Defizite auf. Dazu gehören eine vielfach anzutreffende Dominanz des Abarbeitens von Kalkülen durch die Schüler sowie eine Armut an geometrischen Formen, deren Ursache in der weitgehenden Beschränkung auf lineare Objekte (Geraden und Ebenen) besteht. Der Unterricht wird von Schülern dadurch oftmals als uninteressant empfunden.
- Wesentliche Grundlagen der 3D-Computergrafik basieren auf den im Stoffgebiet Analytische Geometrie zu behandelnden Inhalten und können auf deren Grundlage verständlich werden; andererseits besteht die Möglichkeit, den Unterricht in analytischer Geometrie durch die Nutzung von 3D-Grafiksoftware anschaulicher zu gestalten, gewissermaßen zu „geometrisieren“.
- Die Anfertigung von dreidimensionalen Computergrafiken und -animationen mithilfe geeigneter Software kann Schüler motivieren, sich mit dafür benötigten Inhalten der analytischen Geometrie zu beschäftigen. Durch die Bearbeitung verhältnismäßig offen gestellter Aufgaben aus diesem Bereich ergeben sich Chancen, in stärkerem Maße schülerbezogene Arbeitsformen in den Unterricht einzubeziehen.

---

<sup>1</sup>Im Folgenden werden für ‚Schülerinnen und Schüler‘ sowie ‚Lehrerinnen und Lehrer‘ aus Gründen der besseren Lesbarkeit jeweils die abkürzenden Bezeichnungen ‚Schüler‘ bzw. ‚Lehrer‘ verwendet.

- Die Einbeziehung von Elementen der 3D-Computergrafik erleichtert bzw. ermöglicht erst die Herstellung von Bezügen zum Stoffgebiet Analysis sowie zu den Fächern Physik, Informatik und Kunst.

Im Bereich des Computereinsatzes im Mathematikunterricht weist die 3D-Computergrafik die Besonderheit auf, dass sie sowohl als *Unterrichtsgegenstand* (Verstehen der Funktionsweise der Computergrafik auf Grundlage der Begriffe und Methoden der analytischen Geometrie) als auch als *Hilfsmittel* (für die Visualisierung und für experimentelles Arbeiten) von Bedeutung ist.

Grundlagen der in der vorliegenden Arbeit angestellten Überlegungen und unterbreiteten Vorschläge bilden zwei voneinander weitgehend unabhängige Themenbereiche:

- analytische Geometrie als schulischer Lerninhalt: Ziele, Inhalte und fundamentale Ideen; Defizite der gegenwärtigen curricularen Situation und Unterrichtspraxis sowie Vorschläge und Ansätze zu ihrer Weiterentwicklung (Kapitel 1);
- mathematische Grundlagen der 3D-Computergrafik und ihrer Anwendungen; Bezüge zur Funktionsweise und Bedienung verschiedener Softwarekategorien zur Erstellung von 3D-Visualisierungen (Kapitel 2).

Basierend auf diesen beiden Aspekten werden in Kapitel 3 Potenzen und Ziele des Einsatzes dreidimensionaler Grafiksoftware und der Thematisierung mathematischer Grundlagen der Computergrafik im Stoffgebiet Analytische Geometrie herausgearbeitet und es wird die Auswahl für die Nutzung im Unterricht geeigneter Software diskutiert. In Kapitel 4 erfolgt eine Konkretisierung dieser Überlegungen zu Unterrichtsvorschlägen für die Einbeziehung von Elementen der 3D-Computergrafik in das Stoffgebiet anhand einer Reihe von Gegenstandsbereichen. Die dabei entwickelten Einzelvorschläge einbeziehend, werden Gesamtkonzeptionen für den Aufbau des Stoffgebietes in Grund- und Leistungskursen skizziert. Erfahrungen aus Unterrichtsprojekten, die auf der Grundlage einiger der in Kapitel 4 entwickelten Konzepte durchgeführt wurden, sind Gegenstand von Kapitel 5.<sup>2</sup>

Die vorliegende Arbeit entspricht im Wesentlichen der im April 2006 eingereichten Fassung. Neben der Korrektur weniger kleinerer Fehler und der Aktualisierung einiger Internet-Quellenangaben wurde in Einzelfällen auf mittlerweile verfügbare neue Software-Versionen eingegangen.

Zu dieser Arbeit stehen zahlreiche elektronische Materialien (Beispieldateien, Materialien zu Unterrichtsversuchen, Arbeiten von Schülern und Studierenden u. a.) auf der Internetseite <http://www.afiller.de/habil> zur Verfügung (siehe Anhang D, S. 365).

Andreas Filler, Mai 2007

---

<sup>2</sup>Die beschriebenen Unterrichtsversuche stellen lediglich eine Ergänzung der vorliegenden, hauptsächlich theoretisch-konzeptionell angelegten, Arbeit dar und sollen einige der unterbreiteten Vorschläge illustrieren. Sie können schon aufgrund ihres relativ geringen Umfangs nicht als fundierte empirische Untersuchungen angesehen werden, so dass nur eine Darlegung und qualitative Auswertung exemplarisch gesammelter Erfahrungen erfolgt.



# Inhaltsverzeichnis

<b>1</b>	<b>Entwicklung, Ziele und Probleme des Unterrichts im Stoffgebiet Analytische Geometrie</b>	<b>1</b>
1.1	Zur Geschichte der analytischen Geometrie . . . . .	2
1.1.1	Klassische Koordinatengeometrie . . . . .	2
1.1.2	Vektorielle analytische Geometrie . . . . .	4
1.1.3	Axiomatische Fundierung des Vektorraumbegriffs . . . . .	5
1.1.4	Lineare Gleichungssysteme, Determinanten, Matrizen . . . . .	7
1.1.5	Geometrische Abbildungen und Gruppen . . . . .	7
1.1.6	Zusammenfassung . . . . .	9
1.2	Analytische Geometrie im Mathematikunterricht – historischer Überblick	12
1.2.1	Die Meraner Reformen . . . . .	12
1.2.2	Einbeziehung der Vektorrechnung in den Mathematikunterricht	16
1.2.3	Axiomatisch begründete lineare Algebra – „Neue Mathematik“ .	18
1.2.4	Analytische Geometrie und lineare Algebra im Mathematikunterricht nach dem Scheitern der „Neuen Mathematik“ . . . . .	21
1.2.5	Zusammenfassung und einige Schlussfolgerungen . . . . .	23
1.3	Überblick über derzeitige Rahmenpläne für die analytische Geometrie .	25
1.4	Defizite des Unterrichts und Ansätze zu seiner Weiterentwicklung . . .	32
1.4.1	Analytische Geometrie und Allgemeinbildung . . . . .	32
1.4.2	Kalkülorientierung und Dominanz von Routineaufgaben . . . .	34
1.4.3	Grundvorstellungen und fundamentale Ideen . . . . .	36
1.4.4	Anschauung und Raumvorstellung . . . . .	40
1.4.5	Vorschläge zur Überwindung der Formenarmut des Unterrichts in analytischer Geometrie – ein Überblick . . . . .	45
1.4.6	Anwendungsbezug und mathematische Modellierung . . . . .	49
1.4.7	Vernetzungen zwischen analytischer Geometrie und Elementargeometrie sowie Analysis . . . . .	51

1.4.8	Unterrichtskultur und Unterrichtsmethoden . . . . .	54
1.5	Zur Computernutzung im Unterricht der analytischen Geometrie . . . .	56
1.5.1	Computereinsatz im Mathematikunterricht – didaktische Aspekte	56
1.5.2	Computernutzung im Stoffgebiet Analytische Geometrie . . . . .	60
1.6	Zusammenfassung, Zwischenfazit und Ausblick . . . . .	61
1.6.1	Mögliche Entwicklungsrichtungen des Unterrichts in analytischer Geometrie . . . . .	61
1.6.2	Computergrafik im Unterricht der analytischen Geometrie – Überblick über Potenzen und Ziele . . . . .	62
<b>2</b>	<b>Mathematische Grundlagen der 3D-Computergrafik und ihre Umsetzung in Grafiksoftware</b>	<b>65</b>
2.1	Grundlegende Verfahren der Computergrafik und der Bildbearbeitung .	67
2.1.1	Repräsentation und Verarbeitung von Bildern und Grafiken im Computer – Raster- und Vektorgrafik . . . . .	67
2.1.2	Licht und Farben, mathematische Beschreibung von Farben . .	74
2.2	Überblick über die Funktionsweise der 3D-Computergrafik . . . . .	80
2.2.1	Ein einführendes Beispiel . . . . .	80
2.2.2	Arbeitsschritte der 3D-Computergrafik . . . . .	83
2.3	Modellierung geometrischer Objekte im Raum; geometrische Repräsentationen . . . . .	85
2.3.1	Koordinatensysteme; kartesische und homogene Koordinaten . .	85
2.3.2	Körper und Flächen . . . . .	87
2.3.3	Geometrische Grundkörper . . . . .	87
2.3.4	Constructive Solid Geometry (Boolesche Operationen) . . . . .	89
2.3.5	Beschreibung von Flächen durch Funktionsgleichungen, implizite Gleichungen und Parameterdarstellungen . . . . .	90
2.3.6	Bézier- und Splinekurven . . . . .	98
2.3.7	Konstruktion von Flächen und Körpern aus Kurven . . . . .	104
2.3.8	Freiformflächen . . . . .	106
2.3.9	Polygondarstellung von dreidimensionalen Objekten . . . . .	108
2.3.10	Diskretisierung des Raumes (Voxel) . . . . .	112
2.4	Transformationen und Sichtbarkeitsuntersuchungen . . . . .	113
2.4.1	Affine Abbildungen des Raumes auf sich . . . . .	113
2.4.2	Koordinatentransformationen . . . . .	116
2.4.3	Abbildungen in eine Ebene (Parallel- und Zentralprojektionen) .	118

2.4.4	Clipping von Objekten außerhalb des Sichtvolumens . . . . .	119
2.4.5	Entfernung verdeckter Objekte und Facetten . . . . .	121
2.4.6	Texture-Mapping (Projektion von Bildern auf Flächen) . . . . .	124
2.5	Erzeugung fotorealistischer 3D-Darstellungen . . . . .	128
2.5.1	Reflexion, Transparenz und Brechung . . . . .	128
2.5.2	Lokale Beleuchtungsmodelle, Beleuchtungskomponenten, Simulation von Oberflächenstrukturen . . . . .	131
2.5.3	Interpolation von Farben oder Oberflächennormalen . . . . .	136
2.5.4	Scanline Rendering . . . . .	139
2.5.5	Verfolgung von Lichtstrahlen (Raytracing) . . . . .	140
2.5.6	Globale Beleuchtungsmodelle (Radiosity) . . . . .	145
2.5.7	Anti-Aliasing . . . . .	149
2.6	Animationen . . . . .	151
2.6.1	Animationen auf der Grundlage von Parameterdarstellungen . . . . .	151
2.6.2	Zeitabhängige Beschreibung von Transformationen . . . . .	153
2.6.3	Animierte Formen, Farben und Oberflächeneigenschaften . . . . .	155
2.6.4	Keyframe- und Pfadanimationen, komplexere Techniken . . . . .	156
2.6.5	3D-Animationen in Echtzeit, Interaktivität . . . . .	158
2.7	Überblick über Software für die Erstellung dreidimensionaler computergrafischer Darstellungen . . . . .	160
2.7.1	Architektur, Komponenten und Schnittstellen von 3D-Grafiksoftware (Überblick) . . . . .	160
2.7.2	Szenenbeschreibungssprachen und Raytracing-Software . . . . .	162
2.7.3	3D-Grafiksoftware mit grafischen Benutzeroberflächen . . . . .	166
2.7.4	CAD-Systeme . . . . .	171
2.7.5	Computeralgebrasysteme . . . . .	173
2.7.6	Software für den Unterricht in analytischer Geometrie . . . . .	175
2.7.7	DGS und Software für den Unterricht in Raumgeometrie . . . . .	176
<b>3</b>	<b>Grundsätzliche Überlegungen zur Einbeziehung von Elementen der 3D-Computergrafik in das Stoffgebiet Analytische Geometrie</b>	<b>179</b>
3.1	Überblick über Vorschläge zur Einbeziehung von Elementen der Computergrafik in den Mathematikunterricht . . . . .	180
	Exkurs: Kurse zur Einführung in die Computergrafik an Hochschulen . . . . .	186
3.2	Ziele und Herangehensweisen der Einbeziehung der 3D-Computergrafik . . . . .	187
3.2.1	Übergreifende Ziele . . . . .	187

3.2.2	Aspekte und Herangehensweisen . . . . .	189
3.3	Sichtweisen auf zentrale Gegenstände der analytischen Geometrie im Kontext der Einbeziehung von Elementen der Computergrafik . . . . .	191
3.3.1	Zur Einführung und Verwendung von Vektoren . . . . .	191
	– Didaktische Positionen hinsichtlich des Vektorbegriffs . . . . .	191
	– Der Vektorbegriff in der Informatik, speziell der Computergrafik . . . . .	194
	– Schlussfolgerungen . . . . .	196
3.3.2	Sichtweisen auf Parameterdarstellungen . . . . .	200
3.4	Auswahl geeigneter Software für den Unterricht . . . . .	203
3.4.1	Auswahlkriterien . . . . .	203
3.4.2	Eignung verschiedener Softwarekategorien . . . . .	204
3.4.3	Vor- und Nachteile von CAS und Raytracing-Software . . . . .	207
3.5	Vorlagen und Anleitungen für die Nutzung von POV-Ray im Unterricht . . . . .	211
<b>4</b>	<b>Unterrichtsvorschläge und -konzepte zur Einbeziehung von Elementen der 3D-Computergrafik in das Stoffgebiet Analytische Geometrie</b>	<b>215</b>
4.1	Geometrische Modellierung mittels räumlicher Koordinaten als Einstieg in das Stoffgebiet . . . . .	217
4.1.1	Beschreibung einfacher Körper durch räumliche Koordinaten . . . . .	217
4.1.2	Reduktion der zu berücksichtigenden Dimensionen durch die Betrachtung von Schnitten mit Koordinatenebenen . . . . .	220
4.1.3	Beschreibung einfacher geometrischer Objekte durch Gleichungen . . . . .	221
4.1.4	Anwendung von Kreis- und Kugelgleichungen für die Lösung von Positionierungsaufgaben . . . . .	224
4.1.5	Fertigstellung der ersten Grafiken durch die Schüler; Verwendung von Variablen, Berechnungen in POV-Ray . . . . .	225
4.1.6	Fazit, Anmerkungen, Erfahrungen und einige Ergebnisse . . . . .	227
4.1.7	Verwendung eines CAS für den beschriebenen Einstieg . . . . .	230
4.1.8	Mögliche Wege der Weiterführung des Unterrichts . . . . .	231
4.2	Visualisierung einiger Standardinhalte des Stoffgebietes . . . . .	233
4.2.1	Veranschaulichung der Lage von Punkten im Koordinatensystem sowie von Vektoren durch Pfeile . . . . .	234
4.2.2	Geraden im Raum und Lagebeziehungen zwischen ihnen . . . . .	237
4.2.3	Darstellung von Ebenen; Schnittpunkte von Geraden und Ebenen . . . . .	242
4.2.4	Schnittgeraden von Ebenen, Veranschaulichung linearer Gleichungssysteme und des GAUSS-Algorithmus . . . . .	245

4.3	Erstellung von Animationen durch die parameterabhängige Beschreibung von Kurven . . . . .	249
4.3.1	Koordinatenbezogene Parameterdarstellungen von Kurven . . .	249
4.3.2	Vektorielle Parameterdarstellungen . . . . .	252
4.3.3	Bestimmung von Bahngeschwindigkeiten auf gekrümmten Bewegungsbahnen als Möglichkeit der theoretischen Vertiefung . . . .	254
4.4	Geometrische Transformationen . . . . .	255
4.4.1	Stellenwert und Arten der Beschreibung von Transformationen bei der Einbeziehung von Elementen der Computergrafik . . . .	255
4.4.2	Parallel- und Zentralprojektionen . . . . .	256
4.4.3	Translationen, Drehungen und Skalierungen . . . . .	257
4.4.4	Animationen durch zeitabhängige geometrische Transformationen	259
4.5	Visuell gestützte Untersuchung von Kegelschnitten, Flächen 2. Ordnung und weiteren Flächen . . . . .	261
4.5.1	Räumlich-visuelle Einführung der Kegelschnitte . . . . .	261
4.5.2	Flächen zweiter Ordnung . . . . .	265
4.5.3	Flächen als Graphen von Funktionen zweier Variablen . . . . .	269
4.5.4	Zugänge zu Parameterdarstellungen von Flächen . . . . .	270
4.5.5	Zusammenfassung und Einordnung . . . . .	274
4.6	Einbeziehung mathematischer Grundlagen der 3D-Computergrafik bei der Behandlung des Skalarproduktes und von Normalenvektoren . . . .	275
4.6.1	Das Reflexionsgesetz als Ausgangspunkt . . . . .	275
4.6.2	Zur Einführung des Skalarproduktes . . . . .	277
4.6.3	Normalenvektoren von Ebenen . . . . .	279
4.6.4	Aufgaben zum räumlichen Reflexionsgesetz . . . . .	281
4.6.5	Lokale Beleuchtungsmodelle . . . . .	283
4.6.6	Kantenglättung durch Normaleninterpolation . . . . .	286
4.7	Einbeziehung weiterer Inhalte und Anwendungen der Computergrafik .	289
4.8	Zusammenfassende Anmerkungen zur Berücksichtigung von Aspekten der Informatik . . . . .	290
4.9	Varianten für den Aufbau des Stoffgebietes Analytische Geometrie unter Einbeziehung von Elementen der Computergrafik . . . . .	292
4.9.1	Lehrgangsvarianten für Grundkurse . . . . .	293
4.9.2	Einbeziehung von Elementen der 3D-Computergrafik in den Unterricht von Leistungskursen . . . . .	296
	– Bezüge zum Unterricht in Analysis . . . . .	298

<b>5 Unterrichtserfahrungen hinsichtlich der Einbeziehung von Elementen der 3D-Computergrafik</b>	<b>299</b>
5.1 Unterrichtsversuch in einem Grundkurs am Andreas-Gymnasium, Berlin	300
5.1.1 Rahmenbedingungen, Voraussetzungen der Schüler . . . . .	300
5.1.2 Planung und Verlauf der Unterrichtsreihe . . . . .	302
5.1.3 Abschlussbefragung der Schüler zur Einbeziehung von Elementen der 3D-Computergrafik . . . . .	309
5.1.4 Zusammenfassung und einige Schlussfolgerungen . . . . .	312
5.2 Unterrichtsversuch an der Internationalen Gesamtschule Heidelberg . .	314
5.3 Unterrichtsprojekt am Fürst-Pückler-Gymnasium, Cottbus . . . . .	318
5.4 Erfahrungen aus Seminaren mit Studierenden . . . . .	322
5.5 Abschließende Bemerkungen . . . . .	324
<b>A Anleitungen für Schüler, Vorlagen und Bibliotheken</b>	<b>329</b>
Kurzanleitung für die Arbeit mit POV-Ray . . . . .	330
Kurzanleitung: Transformationen, Boolesche Operationen, Prismen und Rotationskörper in POV-Ray . . . . .	332
Kurzanleitung: Darstellung von Punkten, Vektoren, Geraden und Ebenen . .	335
Kurzanleitung: Darstellung von Flächen, die durch Gleichungen oder Parameterdarstellungen gegeben sind (isosurface, parametric) . . . . .	337
Kurzanleitung: Mit POV-Ray Videos erstellen . . . . .	339
Vorbereitete Objekte für den Einstieg in POV-Ray (vorlage.inc) . . . . .	341
Makros für die Darstellung von Punkten, Strecken, Pfeilen, Geraden, Ebenen	343
<b>B Farbige Abbildungen</b>	<b>349</b>
Abbildungen zu Kapitel 2 . . . . .	349
Von Schülern angefertigte Abbildungen . . . . .	357
Nachweis der Bildquellen . . . . .	363
<b>C Notation</b>	<b>364</b>
<b>D Hinweise zu den Dateien auf der Internetseite zu dieser Arbeit</b>	<b>365</b>
<b>Literaturverzeichnis</b>	<b>367</b>
<b>Stichwortverzeichnis</b>	<b>385</b>

# Kapitel 1

## Entwicklung, Ziele und Probleme des Unterrichts im Stoffgebiet Analytische Geometrie

Eine Analyse der analytischen Geometrie als schulischer Lerninhalt bildet eine wesentliche Grundlage dieser Arbeit. Deshalb werden in diesem Kapitel folgende Themengebiete erörtert:

- Gründe für die Einbeziehung der analytischen Geometrie in den Mathematikunterricht und dabei verfolgte Ziele; historische Entwicklung dieses Stoffgebietes;
- Analyse der gegenwärtigen Curricula und der in den letzten Jahrzehnten erfolgten Veränderungen hinsichtlich der Stoffauswahl und der Zielsetzungen;
- Probleme und Defizite des gegenwärtigen Unterrichts in analytischer Geometrie; Auswertung aktueller Analysen und Vorschläge zu dieser Thematik;
- Grundvorstellungen, die für die analytische Geometrie fundamental sind und welche die Schüler deshalb erwerben sollten;
- Implikationen neuer Anwendungen der analytischen Geometrie für den Unterricht.

In Auswertung dieser Analysen erfolgt eine Zusammenfassung möglicher Perspektiven und Entwicklungsrichtungen des Unterrichts in analytischer Geometrie. Anhand der dabei angestrebten Ziele werden in den Kapiteln 3 und 4 Potenzen und Chancen der Einbeziehung von Elementen der Computergrafik erörtert sowie entsprechende Vorschläge entwickelt.<sup>1</sup>

Da die Computernutzung ein zentraler Bestandteil des Themas dieser Arbeit ist, enthält das Kapitel einen Überblick über Vorschläge zur Einbeziehung des Computers in den Mathematikunterricht der S II, insbesondere in das Stoffgebiet Analytische Geometrie.

---

<sup>1</sup>Didaktische Einzelfragen zur Einführung des Vektorbegriffs, zur Erarbeitung von Parameterdarstellungen sowie zur Behandlung des Skalarproduktes sind ebenfalls Gegenstände der Kapitel 3 und 4, da hierfür auch Konsequenzen aus der Verwendung und Repräsentation dieser Inhalte in der Computergrafik zu ziehen sind, deren mathematische Grundlagen in Kapitel 2 behandelt werden.

## 1.1 Zur Geschichte der analytischen Geometrie

Die historische Entwicklung der analytischen Geometrie und die ihrer Einbeziehung in den Mathematikunterricht stellen wichtige Grundlagen der vorliegenden Arbeit dar. Die Herausbildung der analytischen Geometrie als Teilgebiet der Mathematik kann zu Schlussfolgerungen hinsichtlich von Erkenntnisprozessen in diesem Bereich führen, die (wenn auch nicht zwingend) didaktische Relevanz besitzen. Außerdem ergeben sich Aufschlüsse über das Verhältnis von analytischer Geometrie und linearer Algebra, das in der didaktischen Diskussion von Bedeutung ist. Da u. a. FRANK in [81] (1983) sowie WITTMANN in [247] (2000) und [276] (2003) die Geschichte der analytischen Geometrie recht eingehend dargestellt und auf – auch für die vorliegende Arbeit relevante – didaktische Schlussfolgerungen hin analysiert haben, wird hier lediglich eine Zusammenfassung unter Ergänzung einiger Aspekte gegeben, die m. E. bedeutsam sind.

Als Gegenstand der analytischen Geometrie kann in allgemeinem Sinne die Untersuchung geometrischer Probleme mit rechnerischen (algebraischen) Methoden angesehen werden.<sup>2</sup> Die Grundlage dafür besteht darin, geometrischen Objekten Zahlen (Koordinaten) zuzuordnen und umgekehrt. Geometrische Konstruktionen und Untersuchungen lassen sich dann durch Verfahren der Algebra ersetzen bzw. mittels dieser durchführen. Damit stellt die analytische Geometrie mächtige Hilfsmittel zur Lösung geometrischer Probleme sowie für die Physik und andere Naturwissenschaften zur Verfügung. Umgekehrt können in bestimmten Fällen auch algebraische Fragestellungen durch Rückführung auf geometrische Aufgaben gelöst werden.

Bereits anhand dieser kurzen und recht allgemeinen Bestimmung des Begriffs „Analytische Geometrie“ sei auf ihre Bedeutung als „Bindeglied“ zwischen Geometrie und Computer verwiesen: Alle grafischen Darstellungen und geometrischen Konstruktionen mithilfe des Computers erfordern analytische Beschreibungen der auftretenden geometrischen Objekte durch Koordinaten bzw. Gleichungen.<sup>3</sup>

### 1.1.1 Klassische Koordinatengeometrie

Als Begründer der Koordinatenmethode und somit der analytischen Geometrie werden meist RENÉ DESCARTES (1596–1650) und PIERRE DE FERMAT (1607–1665) angesehen, wenngleich Ansätze hierfür bereits früher zu finden waren (siehe u. a. [81], S. 2–6; [237], S. 300–307; [276], S. 18–21 sowie [221]). Besonderen Einfluss erlangte das Werk „La Géométrie“ (1637) von DESCARTES.<sup>4</sup> Sein Leitmotiv für die Verwendung von Koordinaten war die *Beschreibung und Klassifikation ebener Kurven durch Gleichungen*, welche

---

<sup>2</sup>Für eine kurze Begriffsbestimmung vgl. [67], S. 72–75; zu zentralen erkenntnistheoretischen Aspekten im Zusammenhang mit der analytischen Geometrie siehe [253], insbesondere S. 19–32.

<sup>3</sup>Durch die Bedienung mittels grafischer Benutzeroberflächen ist die Verwendung analytischer Methoden für Anwender von Zeichenprogrammen oder dynamischer Geometriesoftware (DGS) oftmals nicht sichtbar, auch wenn diese natürlich software-intern zum Einsatz kommen.

<sup>4</sup>Allerdings geht auf DESCARTES nicht das nach ihm benannte kartesische Koordinatensystem zurück. Er verwendete Bezugssysteme aus i. Allg. nicht zueinander senkrechten Geraden. Zudem traten negative Koordinatenwerte erst wesentlich später bei WALLIS und NEWTON auf (vgl. [276], S. 18f.).



auch noch über einen längeren Zeitraum das Hauptanwendungsgebiet der analytischen Geometrie blieb.<sup>5</sup> Die Entwicklung der Koordinatenmethode war eng mit der intensiven Entwicklung der Astronomie, der Mechanik und anderer Bereiche der Physik und Technik im 17. Jahrhundert verbunden;<sup>6</sup> sie war grundlegend für die Entwicklung der Analysis in der zweiten Hälfte des 17. und im Verlaufe des 18. Jahrhunderts (NEWTON, LEIBNIZ), die ihrerseits zum Aufschwung der analytischen Geometrie beitrug.

Eine bedeutende Weiterentwicklung der analytischen Geometrie erfolgte durch EULER (1707–1783), der zunächst vor allem das Ziel verfolgte, eine vollständige algebraische Theorie der Kurven zweiten Grades zu entwerfen, und auf den die heutige Ausformung der analytischen Geometrie wesentlich zurückgeht. So führte EULER in seinem Lehrbuch „Introductio in analysin infinitorum“ (1743, 1748) u. a. *Parameterdarstellungen für Kurven* ein, untersuchte systematisch die Kurven 3. und 4. Ordnung und befasste sich mit transzendenten Kurven (u. a. den Graphen von Logarithmusfunktionen und trigonometrischen Funktionen), Zykloiden sowie Spiralen (vgl. u. a. [81], S. 6f. und [237], S. 311ff.). Neben recht- und schiefwinkligen Koordinaten verwendete er auch Polarkoordinaten, die zwar bereits in der 2. Hälfte des 17. Jahrhunderts vereinzelt anzutreffen waren, durch EULERS Arbeiten jedoch erst allgemein bekannt wurden ([58], S. 210).

Die Koordinatenmethode kam zunächst nur für ebene geometrische Objekte zum Einsatz; die Beschreibung des dreidimensionalen Raumes durch Koordinaten erfolgte ansatzweise zuerst durch LA HIRE (1679) und BERNOULLI (1697); systematisch verwendet wurden *räumliche Koordinaten* erstmals 1731 von CLAIRAUT (vgl. [81], S. 7). Die Verallgemeinerung auf *n-dimensionale Räume* wurde erst im 19. Jahrhundert u. a. durch CAYLEY und JORDAN vorgenommen (vgl. [276], S. 21), wobei bereits vektorielle Herangehensweisen zum Einsatz kamen, auf deren Herausbildung noch einzugehen ist. Vorbereitet wurde der Übergang zum *n-dimensionalen Raum* jedoch bereits durch LAGRANGE, der in seiner „Analytischen Mechanik“ (1788) Zustands- bzw. Phasenräume mechanischer Systeme betrachtet. In diesem Zusammenhang wurde auch erstmals die analytische Behandlung linearer Probleme interessant, da synthetische Vorgehensweisen nicht mehr angewendet werden konnten; „elementargeometrische“ Fragestellungen begannen Eingang in die analytische Geometrie zu finden.<sup>7</sup>

Während die Entwicklung der „klassischen“ (vektorfreen) Koordinatengeometrie gegen Ende des 18. Jahrhunderts einen relativen Abschluss gefunden hatte,<sup>8</sup> der sich auch in

---

<sup>5</sup> Im Mittelpunkt des Interesses standen dabei zunächst die Kegelschnitte (als algebraische Kurven 2. Ordnung), die bereits von FERMAT und in der Folgezeit u. a. von WALLIS (1616–1703) untersucht wurden. Bereits 1704 veröffentlichte dann NEWTON eine Arbeit zu den Kurven 3. Ordnung.

<sup>6</sup> Dies kommt z. B. in einer Äußerung von NEWTON (1643–1727) zum Ausdruck, der sagte, er sei in der Lage gewesen, „weit zu blicken“, da er „auf den Schultern eines Giganten“ stand und damit WALLIS meinte. Neben Untersuchungen zu unendlichen Reihen veröffentlichte WALLIS bedeutende Abhandlungen zur analytischen Geometrie, u. a. zur Lehre von den Kegelschnitten.

<sup>7</sup> LAGRANGE berechnete bereits 1773 auf analytischem Wege die Flächeninhalte der Seitenflächen einer dreiseitigen Pyramide und bestimmte die der Pyramide einbeschriebene Kugel, vgl. [81], S. 9.

<sup>8</sup> Der Koordinatenbegriff erfuhr allerdings noch eine Verallgemeinerung, als MÖBIUS und PLÜCKER

der Entstehung von Lehrbüchern niederschlug,<sup>9</sup> wurde die Entwicklung der analytischen Geometrie im 19. Jahrhundert insbesondere durch das Eindringen des Vektorbegriffs und die Integration der Matrizenrechnung geprägt.

### 1.1.2 Vektorielle analytische Geometrie

Obwohl die Entwicklung der Vektorrechnung meist mit HAMILTON (1805–1865) und GRASSMANN (1809–1877) in Verbindung gebracht wird,<sup>10</sup> reichen diesbezügliche Ansätze wesentlich weiter zurück. Bereits aus der Antike sind Parallelogrammkonstruktionen zur Ermittlung resultierender Bewegungen überliefert; im 16. und 17. Jahrhundert verwendeten DA VINCI und GALILEI gerichtete Strecken zur Veranschaulichung von Kräften in der Physik. STEVIN (1548–1620) zerlegte Kräfte in Komponenten und nutzte dazu das Kräfteparallelogramm. Bereits 1679 berichtete LEIBNIZ in einem Brief an HUYGENS über Ideen, einen geometrischen Kalkül zu schaffen. Diese Ideen enthielten bereits Ansätze der heutigen Vektorrechnung, wiesen allerdings Defizite auf, wie das Fehlen des Konzepts der gerichteten Strecken (siehe [276], S. 25f.). Interessant sind vor allem die Motive, die LEIBNIZ bewegten, die Entwicklung eines effizienten Kalküls in Angriff zu nehmen, der es ermöglicht, unmittelbar mit geometrischen Objekten zu rechnen: Die Verwendung von Koordinaten führt zu geometrischen Berechnungen, die von der Lage der Objekte abhängen und daher in vielen Fällen unnötig kompliziert sind. In dem bereits erwähnten Brief an HUYGENS schrieb LEIBNIZ (zit. nach [81], S. 17):

*„Ich denke, dass wir noch eine andere Analysis brauchen, eine eigentlich geometrische oder lineare, die uns die Möglichkeit geben würde, die Lage unmittelbar auszudrücken ähnlich dem, wie man mit Hilfe der Algebra eine Größe ausdrückt.“*

Zu den Wegbereitern der Vektorrechnung gehörten im Folgenden vor allem WESSEL (1745–1813) und MÖBIUS (1790–1868), vgl. [81], S. 17ff. und [276], S. 24ff. Neben den bereits erwähnten Motiven der Untersuchung gerichteter Größen in der Physik und dem Rechnen mit Strecken in der Geometrie gab die geometrische Interpretation der komplexen Zahlen Impulse für die Herausbildung des Vektorkalküls.<sup>11</sup> Mit dem Ziel der Verallgemeinerung der komplexen Zahlen für die Beschreibung der Geometrie des drei-

---

um 1830 im Zusammenhang mit Arbeiten auf dem Gebiet der projektiven Geometrie homogene Koordinaten einführten. Natürlich war damit (durch die Beschreibung unendlich ferner Objekte) auch eine Erweiterung des Raumbegriffs verbunden.

<sup>9</sup>Ein für lange Zeit sehr einflussreiches Lehrbuch der analytischen Geometrie wurde 1798/99 von LACROIX veröffentlicht, siehe S. 12. Sogar die Einführung der Bezeichnung „Analytische Geometrie“ wird oft LACROIX zugeschrieben, vgl. [81], S. 10 und [142], S. 106, wobei allerdings nach [58], S. 209 bereits NEWTON in einer seiner Schriften den Namen „Geometria analytica“ für die Koordinatengeometrie verwendete. Umfangreiche Lehrbücher zur Anwendung der Algebra und der Analysis auf die Geometrie stammen auch von MONGE („Application de l’algèbre à la géométrie“, 1805 und „Application de l’analyse à la géométrie“, 4. Aufl. 1819), der sich hauptsächlich mit der Ausbildung von Ingenieuren beschäftigte. Mit den genannten Werken schuf MONGE bereits Ansätze der Differenzialgeometrie.

<sup>10</sup>Das Wort „Vektor“ trat wahrscheinlich erstmals bei HAMILTON auf.

<sup>11</sup>Die komplexen Zahlen wurden bereits im 16. Jahrhundert eingeführt; ihre geometrische Interpretation war Gegenstand einiger Arbeiten zu Beginn des 19. Jh., vgl. [237], S. 409f. und [276], S. 31f.

dimensionalen Raumes entwickelte HAMILTON 1843 den Kalkül der *Quaternionen*, aus dem heraus sich in der 2. Hälfte des 19. Jahrhunderts die Vektoranalysis entwickelte, welche schnell bedeutsame Anwendungen in der Physik, insbesondere in der Elektrodynamik fand. Trotz des großen Einflusses der Arbeiten HAMILTONS können diese aus heutiger Sicht als Umweg bei der Herausbildung der Vektorrechnung angesehen werden, da eine Verallgemeinerung auf Dimensionen, die größer als drei sind, nicht möglich ist.

Annähernd zeitgleich mit HAMILTON befasste sich GRASSMANN mit der Entwicklung eines vektoriellen Systems, verfolgte dabei jedoch einen allgemeineren Ansatz, der von vornherein auf die Beschreibung  $n$ -dimensionaler Räume abzielte. Bereits in seiner (unveröffentlichten) Arbeit „Die Theorie von Ebbe und Flut“ (1840) sind alle wesentlichen Elemente der heutigen Vektorrechnung, einschließlich eines skalaren und eines vektoriellen Produktes, enthalten (vgl. u. a. [237], S. 410ff. und [276], S. 36ff.). Den Begriff „Vektor“ verwendete GRASSMANN nicht; in seinem Hauptwerk „Die lineale Ausdehnungslehre“ (1844) findet sich dafür die Bezeichnung „extensive Größe“. In diesem Werk konstruierte er als „Ausdehnungsgebilde“ bezeichnete Strukturen, die reellen Vektorräumen entsprechen. Diese wurden bei ihm zunächst noch nicht axiomatisch beschrieben, sondern auf induktivem Wege synthetisch erzeugt: Durch eine stetige Änderung eines Elements entsteht ein Ausdehnungsgebilde erster Stufe; durch stetige Änderung eines Ausdehnungsgebildes  $(n-1)$ -ter Stufe ein solches  $n$ -ter Stufe. In der Überarbeitung der „Linealen Ausdehnungslehre“ (1862) konstruierte GRASSMANN  $n$ -dimensionale reelle Vektorräume bereits als Mengen aller Linearkombinationen, die sich aus  $n$  Elementen bilden lassen, und bewies 11 Fundamentalformeln, die im Wesentlichen die heute allgemein verwendeten Vektorraumaxiome beinhalten (vgl. [2], S. 414ff. und [142], S. 145ff.).

Bereits in der Fassung von 1844 enthielt GRASSMANNs Werk fast alle grundlegenden Elemente vektorieller Systeme, wie lineare Abhängigkeit und Unabhängigkeit, den Austauschsatz für Basiselemente, die Invarianz der Dimension sowie die Dimensionsformel für den Durchschnitt und die lineare Hülle der Vereinigung zweier Unterräume. Allerdings wurden seine Arbeiten zunächst kaum zur Kenntnis genommen.<sup>12</sup>

### 1.1.3 Axiomatische Fundierung des Vektorraumbegriffs

PEANO (1858–1932), der einen wesentlichen Beitrag zur axiomatischen Fundierung der Mathematik leistete, griff in seinem Buch „Calcolo geometrico secondo l’Ausdehnungslehre di H. Grassmann“ (1888) GRASSMANNs Arbeiten auf und entwickelte sie weiter. Zwar beschränkte er sich weitgehend auf dreidimensionale Räume, gab jedoch in dem abschließenden Kapitel die erste axiomatische Definition der (endlich- und unendlich-dimensionalen) reellen Vektorräume an und betrachtete lineare Abbildungen eines Vek-

<sup>12</sup>Zu den oft genannten Ursachen gehört der für die damalige Zeit recht hohe Abstraktionsgrad und die schwere Lesbarkeit insbesondere der ersten Fassung. In der Physik etablierten sich die Ideen der Vektorrechnung allerdings wesentlich schneller als in der Mathematik. Unter Physikern entbrannte nach 1870 sogar ein Richtungsstreit zwischen Anhängern der HAMILTONschen und denen der GRASSMANNschen Herangehensweise. Interessanterweise gehörte HAMILTON zu den Ersten, die sich (1853) zu GRASSMANNs Arbeiten äußerten, vgl. [237], S. 412ff.

torraumes in einen anderen (vgl. [142], S. 155ff. und [276], S. 40ff.). PEANOs Arbeiten zur Vektorraumaxiomatik erlangten jedoch keinen nennenswerten Einfluss.

Ein weiteres Axiomensystem wurde 1918 von WEYL in seinem Werk „Raum, Zeit und Materie“ veröffentlicht, der als Ausgangspunkt die Vorstellung von Vektoren als Translationen im Raum heranzog. Sein Axiomensystem ist dem von PEANO ähnlich, enthält jedoch ein „Dimensionsaxiom“, das nur endlich-dimensionale Vektorräume zulässt. In einem zusätzlichen Axiomensystem definierte WEYL die Beziehungen zwischen Vektoren und Punkten (vgl. [142], S. 167ff. und [276], S. 42f.) und begründete somit den affinen Punktraum. Das Skalarprodukt zweier Vektoren beschrieb er in einem „metrischen Axiom“ als Bilinearform und entwickelte durch diese Ergänzung eine Axiomatik des euklidischen Vektor- und Punktraumes (siehe [81], S. 25).

Wesentliche Impulse zur Entwicklung eines axiomatischen Vektorraumbegriffs kamen aus der Funktionalanalysis, die sich mit Funktionenräumen und darauf definierten Funktionalen (linearen Abbildungen in die Skalkörper  $\mathbb{R}$  oder  $\mathbb{C}$ ) beschäftigt.<sup>13</sup> Mitte der dreißiger Jahre setzte sich die im Zusammenhang mit der Funktionalanalysis herausgebildete Vektorraumaxiomatik auch in der linearen Algebra in breitem Umfang durch.<sup>14</sup>

Es sei an dieser Stelle darauf verwiesen, dass die Axiomatisierung des Vektorraumbegriffs eine neue Sichtweise auf bereits bekannte Vektorrepräsentationen implizierte: Räume von  $n$ -Tupeln, Klassen gerichteter Strecken, Verschiebungen, Kräften oder „extensiven Größen“ (bei GRASSMANN) bilden konkrete *Modelle* für Vektorräume, welche die in den Vektorraumaxiomen geforderten und die daraus abgeleiteten Eigenschaften gemeinsam haben. Analog dazu führte die Axiomatisierung des affinen Punktraumes zu einer erweiterten Auffassung des Begriffs „Punkt“ als Element eines affinen Punktraumes. Ein solcher ist z. B.  $\mathbb{R}^n$ , womit ein  $n$ -Tupel reeller Zahlen als Punkt aufgefasst werden kann. Somit müssen Koordinatentupel und Lösungsmengen von Gleichungen nicht mehr als Beschreibung realer geometrischer Objekte betrachtet, sondern können selbst als geometrische Objekte aufgefasst werden.

Gegen Ende des 19. und Anfang des 20. Jahrhunderts wurden viele Bereiche der Mathematik auf gesicherte logische Grundlagen gestellt. So gab HILBERT 1899 ein vollständiges und logisch exaktes Axiomensystem der euklidischen Geometrie an. Ein dreidimensionaler affiner Punktraum mit einem Skalarprodukt über dem zugehörigen Vektorraum erfüllt bei geeigneter Definition der im Hilbertschen Axiomensystem auftretenden Grundbegriffe (Geraden und Ebenen als affine Unterräume der Dimensionen 1 bzw. 2) alle Axiome der euklidischen Geometrie und wird deshalb als *euklidischer Raum* bezeichnet. Auch ohne den verallgemeinernden Strukturbegriff des affinen Raumes lassen sich  $\mathbb{R}^2$  und  $\mathbb{R}^3$  als Modelle von Axiomensystemen der euklidischen Geometrie ansehen.

---

<sup>13</sup>Ohne hier näher auf die Funktionalanalysis einzugehen, sei erwähnt, dass Funktionenräume i. Allg. unendlich-dimensional und koordinatenfrei sind und dadurch eine allgemeine, axiomatische Fundierung des Vektorraumbegriffs erfordern.

<sup>14</sup>Vgl. [84], Bd. 2, S. 392; zusammenfassende Darstellungen zum Einfluss der Funktionalanalysis auf die Entwicklung der Vektorraumaxiomatik enthalten [81], S. 23ff. und [276], S. 43ff.

### 1.1.4 Lineare Gleichungssysteme, Determinanten, Matrizen

Bei den bisherigen Betrachtungen wurde der Beitrag der linearen Algebra zur Entwicklung der analytischen Geometrie weitgehend außer Acht gelassen. Tatsächlich entwickelten sich beide Gebiete über lange Zeit recht unabhängig voneinander, wenngleich Querverbindungen an verschiedenen Stellen auftraten.

*Lineare Gleichungssysteme* (LGS) wurden bereits im Altertum mithilfe verschiedener Verfahren gelöst, u. a. wurde auch die Idee des GAUSS-Algorithmus bereits verwendet.<sup>15</sup> CRAMER entwickelte 1750 eine (heute unter seinem Namen bekannte) Methode für die Lösung von LGS mit 2, 3 oder 4 Unbekannten und ebenso vielen Gleichungen, wobei er die Lösungen als Quotienten von Polynomen in den Koeffizienten des zu lösenden LGS angab. Diese Polynome werden heute (verallgemeinert) *Determinanten* genannt.<sup>16</sup> Ab 1770 untersuchten VANDERMONDE und LAPLACE die Eigenschaften von Determinanten für kleine Anzahlen von Koeffizienten und führten die Entwicklung nach Zeilen oder Spalten ein (LAPLACEScher Entwicklungssatz). CAUCHY wies 1812 die grundlegenden Eigenschaften von Determinanten allgemein nach, brachte die Theorie der Determinanten in eine deduktive Form und führte die Schreibweise in Form quadratischer Schemata unter Verwendung doppelter Indizes ein (vgl. [2], S. 338f.).

Aus heutiger Sicht erstaunlich erscheint die Tatsache, dass sich das Rechnen mit *Matrizen* erst deutlich später entwickelte als der Determinantenkalkül. Obwohl sich Vorformen der Matrizenrechnung bereits bei EULER und GAUSS finden lassen und CAUCHY für seine Theorie der Determinanten bereits eine Schreibweise verwendete, die der heutigen Matrizen Schreibweise recht ähnlich ist, bildete sich der Matrizenkalkül erst um 1850 heraus, wobei u. a. EISENSTEIN, SYLVESTER und insbesondere CAYLEY wichtige Beiträge leisteten. Einige bekannte Anwendungen von Matrizen in der analytischen Geometrie traten jedoch bereits weitaus früher auf. So beschäftigten sich schon EULER und LAGRANGE mit der Diagonalisierbarkeit symmetrischer Matrizen; ein wichtiges Motiv hierfür war die Durchführung von Hauptachsentransformationen quadratischer Formen. Die Theorie der Eigenwerte wurde dann von CAUCHY weiterentwickelt (1829) und als Verallgemeinerung der Diagonalform entwickelte JORDAN 1870/71 die nach ihm benannte Normalform beliebiger quadratischer Matrizen mit komplexen Koeffizienten (siehe [2], S. 400ff. und [276], S. 23f.). Die Entwicklung des Vektorraumbegriffs ermöglichte schließlich zu Beginn des 20. Jahrhunderts eine Neufassung der Matrizentheorie und es bildete sich die Theorie der Eigenräume heraus.

### 1.1.5 Geometrische Abbildungen und Gruppen

Geometrische Abbildungen wurden recht lange nur implizit und im Zusammenhang mit Anwendungen betrachtet (siehe [237], S. 417). Dennoch lassen sich Bezüge zwischen

<sup>15</sup>Auf die vielfältigen Ansätze zur Lösung linearer Gleichungssysteme, die im Verlaufe der Jahrhunderte verfolgt wurden, wird hier nicht eingegangen, siehe dazu z. B. [2], S. 94, 122, 335ff., 400.

<sup>16</sup>Die Bezeichnung „Determinante“ führte GAUSS 1801 ein, der allerdings das nach ihm benannte Lösungsverfahren für LGS gegenüber dem Determinantenkalkül bevorzugte.

der Beschreibung geometrischer Transformationen und der analytischen Geometrie sowie der linearen Algebra – sowohl historisch als auch inhaltlich – in unterschiedlichen Zusammenhängen herstellen.<sup>17</sup> Recht frühe Ansätze der analytischen Beschreibung von Bewegungen gehen auf MÖBIUS (1838) und CLIFFORD (1873) zurück, wobei Letzterer dazu an HAMILTONS Quaternionenkalkül anknüpfte. MÖBIUS, der in [237] als „Vater des abbildungs- und gruppentheoretischen Denkens in der Geometrie“ bezeichnet wird, beschäftigte sich u. a. ausführlich mit projektiven Abbildungen (und den dabei auftretenden Spezialfällen affine, Ähnlichkeits- und Kongruenzabbildungen) sowie sphärischen Bewegungen, Abbildungen der Sphäre in eine Ebene und schließlich (1863) topologischen Abbildungen (von ihm als elementare Verwandtschaften bezeichnet).

Der um 1850 herausgebildete Matrizenkalkül gestattete es, alle linearen Abbildungen durch Matrizen zu beschreiben und ihre Eigenschaften sowie Invarianten algebraisch zu untersuchen. CAYLEY gab 1854 in Verallgemeinerung der zuvor von CAUCHY untersuchten Permutationsgruppen die erste allgemeine Definition des Gruppenbegriffs an. Allerdings erfolgte dieser Abstraktionsschritt recht früh und fand erst ein breiteres Echo, als sich CAYLEY Ende der 70-er Jahre des 19. Jh. erneut dieser Thematik zuwandte (vgl. [2], S. 388ff.). Vorher (1869/70) untersuchte JORDAN diskrete Untergruppen der euklidischen Bewegungsgruppe und stellte somit einen Zusammenhang zwischen den endlichen Gruppen und kontinuierlichen Gruppen geometrischer Abbildungen her. Der Begriff der „Gruppe von Bewegungen“ wurde von JORDAN erstmals verwendet.

1872 legte KLEIN sein berühmt gewordenes Erlanger Programm vor, dessen Grundaussage zu einem ordnenden Prinzip verschiedenster Bereiche der Geometrie wurde:

*„Als Verallgemeinerung der Geometrie entsteht so das folgende umfassende Problem:*

*Es ist eine Mannigfaltigkeit und in derselben eine Transformationsgruppe gegeben; man soll die der Mannigfaltigkeit angehörigen Gebilde hinsichtlich solcher Eigenschaften untersuchen, die durch die Transformationen der Gruppe nicht geändert werden.“<sup>18</sup>*

Obwohl der Begriff der Mannigfaltigkeit nicht mit dem des Vektor- oder des affinen Punktraumes gleichzusetzen ist,<sup>19</sup> lässt sich das Prinzip des Erlanger Programms auf die Untersuchung der Invarianten der Untergruppen der Gruppe der affinen Abbildungen übertragen. Die Untersuchung von Transformationsgruppen auf Vektor- und affinen Punkträumen (bzw. Spezialfällen wie euklidischen Vektor- und Punkträumen) wurde zu einer Herangehensweise an die Untersuchung geometrischer Eigenschaften, die sich in der 2. Hälfte des 20. Jahrhunderts in vielen Hochschulvorlesungen etabliert hat.

---

<sup>17</sup>Ein recht elementarer Zusammenhang ist durch die Beschreibung von Verschiebungen durch Klassen gerichteter Strecken (bzw. Pfeilklassen) gegeben. Auch die schon von EULER und LAGRANGE behandelten Hauptachsentransformationen lassen sich als geometrische Abbildungen deuten, allerdings wurde diese Interpretation damals noch nicht vorgenommen.

<sup>18</sup>Den Begriff „Gruppe“ verwendete KLEIN in der ursprünglichen Fassung seiner als „Erlanger Programm“ bezeichneten Antrittsvorlesung noch nicht, sondern erst in späteren Überarbeitungen.

<sup>19</sup>Eine Mannigfaltigkeit ist ein topologischer Raum, der lokal zu einem Euklidischen Raum (z. B.  $R^n$ ) homöomorph, d. h. auf einen solchen bijektiv und umkehrbar stetig abbildbar ist. Im Ganzen ist dies jedoch i. Allg. nicht der Fall. Ein euklidischer Punktraum ist somit eine sehr spezielle Mannigfaltigkeit.

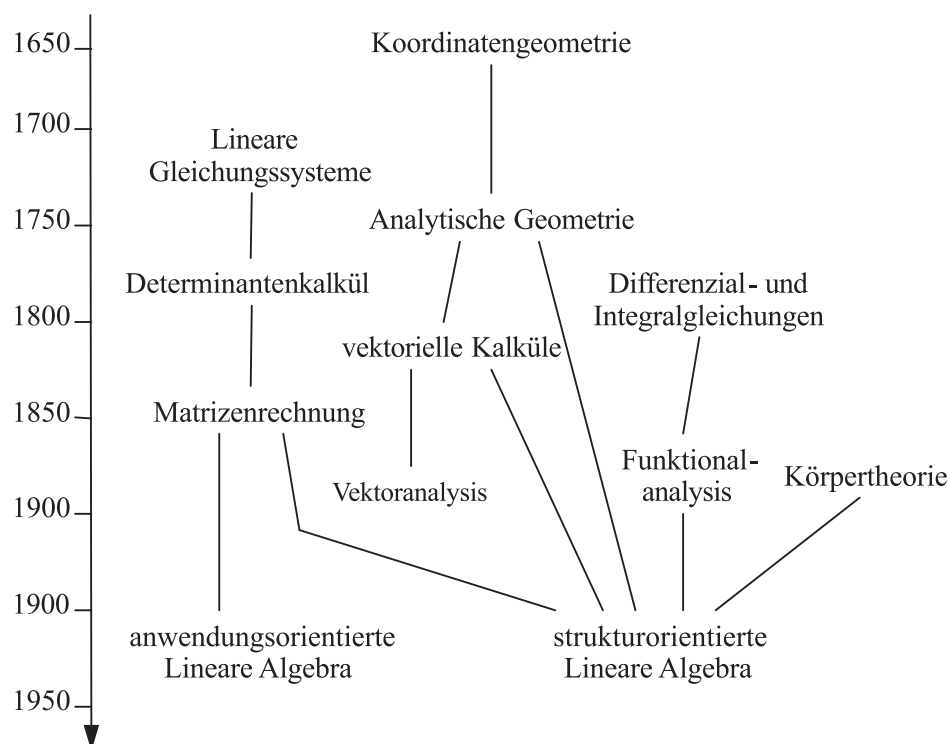


Abbildung 1.1: Entwicklung der analytischen Geometrie und linearen Algebra  
(Abbildung von G. WITTMANN, siehe [247], S. 73 bzw. [276], S. 17)

### 1.1.6 Zusammenfassung

In der ersten Hälfte des 20. Jahrhunderts wurde die Synthese aus analytischer Geometrie, „klassischer“ linearer Algebra sowie Strukturalgebra weitgehend vollendet.<sup>20</sup> Damit hatte einerseits die mit der Herausbildung der klassischen Koordinatengeometrie begonnene „Algebraisierung der Geometrie“ ein sehr hohes und allgemeines Niveau erreicht, andererseits wird im Zusammenhang mit der axiomatischen Begründung algebraischer Strukturen mitunter auch von der „Geometrisierung der Algebra“ gesprochen ([142], S. 107).<sup>21</sup>

Wichtige Entwicklungslinien bei der Herausbildung der heutigen linearen Algebra und analytischen Geometrie verdeutlicht die (unverändert von WITTMANN übernommene) Abbildung 1.1. Natürlich kann ein derartiges Schema nur einen groben Überblick über

<sup>20</sup>Hinsichtlich einiger zu Ende des 19. und in der ersten Hälfte des 20. Jahrhunderts erfolgter Entwicklungen der linearen Algebra unter dem Einfluss der Körper- und Modultheorie siehe [276], S. 46–49.

<sup>21</sup>Allerdings erscheint die Verwendung des Begriffs „Geometrisierung“ in diesem Zusammenhang etwas fragwürdig, ist doch lediglich die Übertragung der in der Geometrie (durch EUKLID) zuerst verwendeten axiomatischen Methode auf die Algebra gemeint. Dabei handelt es sich jedoch nicht um eine genuin geometrische Methode, vielmehr führte die Entwicklung der Mengenlehre und insbesondere der mathematischen Logik zu einer axiomatischen Fundierung unterschiedlicher Gebiete der Mathematik.

Entwicklungsstadien geben und muss eine Reihe von Querverbindungen unberücksichtigt lassen. Ebenfalls unberücksichtigt bleiben in dem Schema die Beziehungen zwischen geometrischen Abbildungen, Matrizen und Gruppen sowie in diesem Zusammenhang die Einbeziehung der Transformationsgruppen in die analytische Geometrie, die schwer terminierbar ist, wenngleich dieser Aspekt aus heutiger Sicht einen wichtigen Inhalt der analytischen Geometrie darstellt.

Für die Herausbildung und „frühe“ Entwicklung der analytischen Geometrie als Koordinatengeometrie kann zusammenfassend festgehalten werden, dass diese aus dem Bedürfnis nach der Beschreibung und Untersuchung recht „anspruchsvoller“ geometrischer Objekte (hauptsächlich Kurven) entstand und Beziehungen zwischen linearen Objekten (Strecken, Geraden, Ebenen) hierbei praktisch keine Bedeutung hatten. Die beiden wichtigsten (eng miteinander verwobenen) Aspekte der frühen Koordinatengeometrie fasste SCHREIBER in [221] folgendermaßen zusammen:<sup>22</sup>

- Die Einführung von Koordinaten ermöglicht es, Beziehungen zwischen geometrischen Objekten in gleichbedeutende Beziehungen zwischen den Koordinaten dieser Objekte zu „übersetzen“. Geometrische Beziehungen lassen sich dadurch arithmetisch-algebraisch ausdrücken.
- Es lassen sich „Bilder vorher vielleicht nie gesehener Kurven machen“, die zunächst nur durch ihre Gleichungen gegeben sind. Die Koordinatenbeschreibung ermöglicht somit geometrische Interpretationen arithmetisch-algebraischer Sachverhalte.

Aus heutiger Sicht ist m. E. ein weiterer Aspekt der Koordinatengeometrie hinzuzufügen:

- Kurven (und auch Flächen), die durch vorgegebene Punkte verlaufen sollen und an die Anforderungen hinsichtlich ihrer Form und bestimmter Eigenschaften (wie z. B. Glattheit sowie Tangentenrichtungen in bestimmten Punkten) gestellt werden, lassen sich aus den Koordinaten vorgegebener Punkte durch Algorithmen generieren, auch wenn geschlossene Darstellungen derartiger Kurven bzw. Flächen durch Gleichungen oder Parameterdarstellungen extrem unübersichtlich wären.

Diese Herangehensweise an die Konstruktion geometrischer Objekte entwickelte sich erst in der 2. Hälfte des 20. Jahrhunderts im Zusammenhang mit dem computerunterstützten Konstruieren (CAD) und der Computergrafik und basiert neben der analytischen Geometrie auch auf der numerischen Mathematik.<sup>23</sup>

WITTMANN zog aus der historischen Entwicklung der linearen Algebra und analytischen Geometrie einige Schlussfolgerungen für die Analyse und Entwicklung von Curricula zur vektoriellen analytischen Geometrie ([276], S. 50):

---

<sup>22</sup>Diese beiden Aspekte sind m. E. auch für den Mathematikunterricht von erheblicher Bedeutung und verdeutlichen vor allem die enge Beziehung zwischen der Untersuchung von Funktionen und der analytischen Geometrie.

<sup>23</sup>Auf algorithmisch erzeugte Kurven und Flächen (Bézierkurven und -flächen, Splines und Subdivision Surfaces) wird im 2. Kapitel dieser Arbeit eingegangen.



- „Die historische Entwicklung der Analytischen Geometrie und Linearen Algebra (zeigt sich) nicht als eine organisch gewachsene Entwicklung, sondern sie zerfällt ... in zahlreiche Entwicklungslinien. Sie verläuft insbesondere deutlich anders als jene der Analysis, für die sich ein vergleichsweise klarer und geradliniger Weg von den Anfängen bis zur Gegenwart ausmachen lässt. In der Analysis erweisen sich deshalb auch historische Sichtweisen und Verfahren als sehr fruchtbar für die Unterrichtspraxis.<sup>24</sup> In der Linearen Algebra und Analytischen Geometrie ist dieser Bezug in der Regel nicht so unmittelbar herzustellen.
- Die vektorielle Analytische Geometrie erwächst aus mehreren Ansätzen, wobei arithmetische oder algebraische und geometrische Sichtweisen in jeweils unterschiedlichen Beziehungen zueinander stehen.
- Ein wichtiger Ursprung der vektoriellen Analytischen Geometrie ist die Suche nach geometrischen Kalkülen. Es entstehen funktionierende Kalküle auf der Basis konkreter geometrischer Objekte, die über lange Zeit hinweg ohne Existenzfragen und ohne strukturelle Überlegungen auskommen - ein Indikator dafür ist die fehlende Rezeption der ersten Axiomatisierungsansätze.
- Die vektorielle Analytische Geometrie wird zwar oftmals zusammen mit der Linearen Algebra in einem Atemzug genannt, nicht zuletzt auch deshalb, weil sie heute kanonische Veranschaulichungen für Begriffe der Linearen Algebra liefert. In der historischen Entwicklung gibt es jedoch keinen direkten Weg vom einen zum anderen Teilgebiet. Die Lineare Algebra ist insbesondere nicht durch eine Abstraktion von Strukturen der vektoriellen Analytischen Geometrie entstanden. Wesentliche Begriffe der Linearen Algebra sind vielmehr aus der Körpertheorie und der Funktionalanalysis erwachsen. Ähnliches gilt auch für Determinanten und Matrizen, deren Strukturen ebenfalls erst sehr spät unter die Begriffe der Linearen Algebra subsumiert werden.“

Diesen, m. E. bedeutsamen, Schlussfolgerungen möchte ich zwei Gedanken hinzufügen, die sich meiner Meinung nach aus der historischen Entwicklung in Bezug auf die Behandlung geometrischer Objekte und Sachverhalte herauskristallisieren:

- Bereits auf der Grundlage der klassischen Koordinatengeometrie wurden anspruchsvolle Untersuchungen interessanter geometrischer Objekte (vor allem Kurven) durchgeführt; siehe hierzu u. a. die Anmerkungen zu den Arbeiten EULERS auf S. 3 und die Inhalte des Lehrbuches von LACROIX, auf das im nächsten Abschnitt noch kurz eingegangen wird.
- Allgemeine Begriffsbildungen und axiomatische Beschreibungen konnten sich erst etablieren, nachdem eine hinreichend große Zahl entsprechender Beispiele bzw. Anwendungen allgemein bekannt war. Sowohl die erste Definition des Gruppenbegriffs durch CAYLEY (1854) als auch die Vektorraumaxiome von PEANO (1888) waren in diesem Sinne „ihrer Zeit voraus“ und wurden zunächst wenig beachtet.

---

<sup>24</sup>WITTMANN bezieht sich hinsichtlich der Geschichte der Analysis auf [252] und verweist auf in [117] dargelegte didaktische Schlussfolgerungen aus der historischen Entwicklung. Zu einer ähnlichen Einschätzung kommt JAHNKE in [123].

## 1.2 Analytische Geometrie und lineare Algebra im Mathematikunterricht – historischer Überblick

Bereits in der zweiten Hälfte des 18. Jahrhunderts fand die klassische Koordinatengeometrie Aufnahme in die Ausbildung an technisch orientierten höheren Lehranstalten. Ein besonders einflussreiches Lehrbuch war dabei „*Traité élémentaire de trigonométrie rectiligne et sphérique et d'application de l'algèbre à la géométrie*“ von LACROIX (1765-1843), das 1798/99 erstmals und 1897 in der 25. Auflage erschien (siehe [81], S. 10f.). In diesem Lehrbuch wird die analytische Geometrie der Ebene, beginnend mit Geradengleichungen, Abstandsberechnungen (Punkt–Punkt sowie Punkt–Gerade) und Berechnungen von Schnittwinkeln von Geraden, behandelt.<sup>25</sup> Es folgen Untersuchungen von Dreiecken und Kreisen sowie anschließend von Ellipsen, Hyperbeln und Parabeln, deren Gleichungen aus der allgemeinen Gleichung 2. Grades hergeleitet werden. Dabei erfolgt auch eine Einführung von Polarkoordinaten und eine Herleitung der gemeinsamen Gleichung der Kegelschnitte in Polarkoordinaten. Anschließend werden Kurven höherer Ordnung betrachtet. LACROIX verwendete in seinem Buch recht moderne Bezeichnungsweisen, die der heutigen Terminologie und Symbolik nahe kommen.

Der kurze Einblick in die Inhalte des Buches von LACROIX könnte den Eindruck eines hoch entwickelten Unterrichts in analytischer Geometrie im 19. Jahrhundert erwecken. Allerdings wurden diese Inhalte nur in berufsspezifischen Bildungseinrichtungen für höhere technische Berufe behandelt. Der Mathematikunterricht allgemeinbildender Schulen (auch der Gymnasien) war davon noch weit entfernt. Bis zur Mitte des 19. Jahrhunderts bestand der Geometrieunterricht an vielen höheren Schulen hauptsächlich im Studium der Bücher 1–6 (Planimetrie) und 11, 12 (Stereometrie, jedoch weitestgehend verkürzt auf Inhaltsberechnungen) der Elemente des EUKLID. Elemente der angewandten Mathematik konnten in Realschulen<sup>26</sup> früher und in größerem Umfang Fuß fassen als in den stärker geisteswissenschaftlich orientierten (humanistischen) Gymnasien.

### 1.2.1 Die Meraner Reformen

In der zweiten Hälfte des 19. Jahrhunderts führte die rasch fortschreitende Industrialisierung zu einem schnell wachsenden Bedarf an hoch qualifizierten technischen Fachkräften, insbesondere Ingenieuren z. B. für Maschinenbau und Elektrotechnik. Der Stellenwert der angewandten Mathematik stieg dadurch enorm und dabei kam der Beschreibung und Lösung geometrischer Sachverhalte und Aufgaben mit arithmetischen und algebraischen Methoden eine hohe Bedeutung zu. Dies führte dazu, dass die ana-

<sup>25</sup>Ab der 4. Auflage (1807) nahm LACROIX dann auch Elemente der analytischen Geometrie des Raumes in das Werk auf und untersuchte die gegenseitige Lage von Punkten, Geraden und Ebenen.

<sup>26</sup>Realschulen entstanden in Deutschland in der Mitte des 18. Jahrhunderts. Es handelte sich dabei bis in das 20. Jh. hinein um recht elitäre Schulen, die (wie auch die Gymnasien) nur von einem geringen Teil der Schüler besucht wurden, vgl. [121], S. 153. Im 19. Jh. bildete sich unter den höheren Schulen eine Differenzierung in humanistische Gymnasien, Realgymnasien, Oberrealschulen und Realschulen heraus. Die überwiegende Mehrzahl der Schüler besuchte jedoch die achtjährige Volksschule.

lytische Geometrie zu Beginn des 20. Jahrhunderts im Zuge der Meraner Reformen obligatorischer Bestandteil des Mathematikunterrichts aller höheren Schulen wurde.<sup>27</sup>

Bereits 1892 fand die Behandlung der Kegelschnitte Eingang in den Lehrplan der preußischen Gymnasien, schon vorher wurden an Oberrealschulen in Preußen Elemente der analytischen Geometrie behandelt (vgl. [81], S. 37 und [160], S. 80). Zur Jahrhundertwende setzte eine koordinierte Reformbewegung für die grundlegende Umgestaltung des Mathematikunterrichts an höheren Schulen ein.<sup>28</sup> Die Zusammenführung aller wichtigen bis dahin herausgearbeiteten Reformvorschläge erfolgte durch die 1904 gegründete Unterrichtskommission der Gesellschaft Deutscher Naturforscher und Ärzte mit den 1905 vorgelegten Meraner Plänen ([99]). Darin heißt es:

*„Einmal gilt es ..., den Lehrgang mehr als bisher dem natürlichen Gange der geistigen Entwicklung anzupassen, überall an den vorhandenen Vorstellungskreis anzuknüpfen, die neuen Kenntnisse mit dem vorhandenen Wissen in organische Verbindung zu setzen, endlich den Zusammenhang des Wissens in sich und mit dem übrigen Bildungsstoff der Schule von Stufe zu Stufe mehr und mehr zu einem bewußten zu machen. Ferner wird es sich darum handeln, unter voller Anerkennung des formalen Bildungswertes der Mathematik doch auf alle einseitigen und praktisch bedeutungslosen Spezialkenntnisse zu verzichten, dagegen die Fähigkeit zur mathematischen Betrachtung der uns umgebenden Erscheinungswelt zu möglicher Entwicklung zu bringen. Von hier aus entspringen zwei Sonderaufgaben: die Stärkung des räumlichen Anschauungsvermögens und die Erziehung zur Gewohnheit des funktionalen Denkens. Die von je dem mathematischen Unterricht zugewiesene Aufgabe der logischen Schulung bleibt dabei unbeeinträchtigt.“*

Die grundlegenden Entwicklungsrichtungen der Meraner Reformen lassen sich somit zu drei Schwerpunkten zusammenfassen:

- Es wurde an vielen Stellen eine im weiten Sinne gefasste *Anwendungsorientierung* des Mathematikunterrichts gefordert.

*„Die Anwendungen sollen auf allen Stufen wirkliche, nicht erdichtete, tatsächlich niemals vorkommende Verhältnisse behandeln; die Aufgaben sollten auch möglichst so ge-*

<sup>27</sup>Die Bedeutung der industriell-technischen Revolution in der zweiten Hälfte des 19. Jahrhunderts für die Reformen des Mathematikunterrichts zur Jahrhundertwende wurde von INHETVEEN in [121] ausführlich diskutiert. Eine Analogie zur Situation ab ca. 1970, als durch das Aufkommen von Informations- und Kommunikationstechnologien radikale Veränderungen von Produktionsprozessen und damit verbundenen Anforderungen an die Qualifikation sowie auch des außerberuflichen Alltags einsetzten, ist unverkennbar. In diesem Sinne formulierten HISCHE und WEIGAND in [112]: *„Wir sind der Ansicht, dass die heutige Situation an der Jahrtausendschwelle – analog zu der Situation vor hundert Jahren – vor dem Phänomen steht, dass sich alle Schulfächer der gesellschaftlichen Bedeutung der Technik und damit auch speziell der Informatik im Rahmen ihres Allgemeinbildungsauftrags stellen müssen.“*

<sup>28</sup>So beschäftigte sich 1897 der 1. Mathematiker-Kongress in Zürich ausführlich mit der Weiterentwicklung des Mathematikunterrichts. Seit 1899 wurde die schulmathematische Zeitschrift „L'Enseignement Mathématique“ herausgegeben, zu deren Redaktionskollegium Mathematiker aus 25 Ländern gehörten. Erwähnt sei hier FELIX KLEIN, der bereits 1872 in seiner Erlanger Antrittsvorlesung die Zerteilung der Hochschulausbildung in eine theoretische und eine technologisch-praktische Richtung kritisierte und sich für eine stärker polytechnisch orientierte Bildung einsetzte. In den folgenden Jahrzehnten wandte er sich Fragen des Schulunterrichts zu und wurde schließlich erster Vorsitzender der 1908 gegründeten Internationalen Mathematischen Unterrichtskommission (IMUK), vgl. [121], S. 132ff.

*stellt sein, daß ihre Ergebnisse wertvoll und von praktischer Bedeutung sind. Bei der Auswahl der Sachgebiete wird auf den Zusammenhang mit anderen Unterrichtsfächern, vor allem aber auf die Berücksichtigung des Lebens- und Anschauungskreises der Schüler und seine zielbewußte Erweiterung zu achten sein, so daß aus dem mathematischen Unterricht auch eine reiche sachliche Belehrung hervorgeht.*“ ([186], S. 77)

Die Befähigung zu einem mathematisierten Umgang mit lebensweltlichen Erscheinungen bis hin zur „*Einsicht in die Bedeutung der Mathematik für die exakte Naturerkenntnis und die moderne Kultur überhaupt*“ ([99], S. 105) erhielt einen hohen Stellenwert innerhalb des Mathematikunterrichts.<sup>29</sup>

- Der Förderung der *Raumanschauung* wurde großes Augenmerk zuteil.<sup>30</sup> Deutlich wird dies u. a. anhand einer Äußerung KLEINS, wonach die sinnliche Anschauung „*überall dem logischen Denken vorausseilt und also in jedem Momente einen weiteren Bereich besitzt als dieses*“ ([121], S. 164). Insbesondere der Geometrieunterricht solle sich an die natürliche Anschauung anschließen und von praktischen Messungen ausgehen. Entsprechend großer Wert wurde auf „geometrisches Zeichnen und Messen“ in allen Klassenstufen gelegt und die Bedeutung von anderen Visualisierungshilfen, wie Modellen für die Raumgeometrie, herausgestellt.<sup>31</sup>
- Die *Erziehung zur Gewohnheit des funktionalen Denkens* erhielt den Rang einer zentralen Aufgabe des Mathematikunterrichts. Erstmals in der Lehrplangeschichte wurden Funktionen in den Mathematiklehrgang der Gymnasien aufgenommen; funktionale Betrachtungen fanden in großem Umfang Eingang in den arithmetischen und algebraischen sowie in den geometrischen Unterricht ([121], S. 194f.). Die Veranschaulichung von Funktionen durch grafische Darstellungen stellte den Zusammenhang mit dem Schwerpunkt der Raumanschauung her.

Im Ergebnis der Meraner Reformen wurden Elemente der Differenzial- und Integralrechnung sowie der analytischen Geometrie (im Sinne klassischer Koordinatengeometrie) feste Bestandteile des Mathematikunterrichts in Deutschland und in Folge der Tätigkeit der IMUK in vielen anderen Ländern (siehe [81], S. 39).

<sup>29</sup>Unschwer lassen sich die knapp 100 Jahre später in den Bildungsstandards der KMK zu findenden Leitideen „Modellieren“ und „Vernetzung“ bereits in den Meraner Plänen erkennen. Auch die Leitideen „Funktionaler Zusammenhang“ sowie „Raum und Form“ finden sich bereits unmittelbar in den Meraner Vorschlägen wieder, so dass es nahe liegt, die zu Anfang des 21. Jahrhunderts stattfindenden Veränderungen des Mathematikunterrichts in Deutschland in inhaltlicher Hinsicht als eine Renaissance der Meraner Reformen, allerdings für einen weiteren Kreis von Schülern, aufzufassen.

<sup>30</sup>Der Begriff Raumanschauung ist dabei weiter aufzufassen als im Hinblick auf das Vermögen der Vorstellung von Objekten und ihrer Lage im dreidimensionalen Raum. Er beinhaltet die Vorstellung geometrischer Beziehungen in Ebene und Raum sowie die Entwicklung geometrisch-anschaulicher Vorstellungen von Größen und Beziehungen, die ihrem Ursprung nach nicht geometrisch sein müssen.

<sup>31</sup>INHETVEEN stellte die große Bedeutung, welche grafischen Darstellungen und Visualisierungen durch die Meraner Lehrpläne zugewiesen wurde, in einen Zusammenhang mit der industriellen Entwicklung in der 2. Hälfte des 19. Jahrhunderts und dem dadurch entstandenen hohen Bedarf an Wissenschaftlern und Konstrukteuren mit fundierten zeichnerisch-konstruktiven Fähigkeiten ([121], S. 192f.). Die Analogie zu der in den letzten Jahrzehnten eminent gestiegenen Bedeutung von computerunterstütztem Entwurf (CAD) und Computergrafik ist aus heutiger Sicht kaum zu übersehen.

Für die beiden höchsten Klassenstufen der humanistischen Gymnasien sahen die Meraner Lehrpläne von 1905 folgende geometrische Unterrichtsinhalte vor (vgl. [99]):

- Stereometrie unter Berücksichtigung von Elementen der Projektionslehre,
- einfachste Sätze der sphärischen Trigonometrie,
- mathematische Geographie einschließlich der Lehre von den Kartenentwürfen,
- Kegelschnitte in analytischer und in synthetischer Behandlung.

Für Realgymnasien und Oberrealschulen enthielten die Meraner Pläne von 1905 keine Empfehlungen.<sup>32</sup> 1922 erschienen die „Revidierten Meraner Lehrpläne des Deutschen Ausschusses für den Mathematischen und Naturwissenschaftlichen Unterricht“ ([186]), welche für Oberrealschulen die Behandlung folgender geometrischer Inhalte vorsahen:<sup>33</sup>

- sphärische Trigonometrie, mathematische Erd- und Himmelskunde,
- analytische Geometrie der Geraden und des Kreises,
- ebene Schnitte durch Zylinder und Kegel, Zusammenhang mit den Ortsdefinitionen der Kegelschnitte, synthetische Geometrie der Kegelschnitte,
- analytische und (in Ansätzen) projektive Behandlung der Kegelschnitte,
- Darstellung von Teilen der Sphäre (Kartenlehre),
- Rückblick auf Grundlagen der Geometrie, philosophische Ausblicke.

Die im Zuge der Meraner Reformen entstandenen Lehrpläne bildeten die Grundlage für den Mathematikunterricht in Deutschland<sup>34</sup> bis zum Ende der fünfziger Jahre des 20. Jahrhunderts (vgl. [81], S. 39 und [247], S. 93). Die analytische Geometrie wurde bis zu diesem Zeitpunkt als Koordinatengeometrie, fast ausschließlich der Ebene, behandelt. Im Mittelpunkt standen die Kegelschnitte, für deren Untersuchung der größte Teil der Unterrichtszeit verwendet wurde. Es bildeten sich dabei „Aufgabeninseln“ (Typen von Routineaufgaben) heraus, für deren Bearbeitung ein großer Teil der Unterrichtszeit verwendet wurde.<sup>35</sup> Die Rolle einer derartigen Aufgabendidaktik wird bei der Analyse der gegenwärtigen Unterrichtspraxis im Stoffgebiet Analytische Geometrie in den Abschnitten 1.4.2 und 1.4.8 noch ausführlicher zur Sprache kommen.

<sup>32</sup>Für die Mathematik und die Naturwissenschaften stand an Oberrealschulen die höchste und an humanistischen Gymnasien die niedrigste Stundenzahl zur Verfügung.

<sup>33</sup> Die Meraner Lehrplanempfehlungen beeinflussten wesentlich die Gestaltung der Lehrprogramme in allen Ländern Deutschlands, wobei einige Länder noch über die in den Meraner Plänen genannten Inhalte hinausgingen. So gehörte die analytische Behandlung der Flächen 2. Ordnung nach den Lehrplänen von 1912 zum Unterrichtsstoff der höchsten Klassenstufe der Oberrealschulen des Königreiches Württemberg (vgl. [88]). Andere Länder legten, zusätzlich zu den in den Meraner Richtlinien enthaltenen Kegelschnitten, die analytische Behandlung verschiedener Kurven fest (vgl. [188]).

<sup>34</sup>Es traten in dieser Hinsicht keine bedeutenden Unterschiede zwischen der Bundesrepublik Deutschland und der DDR auf, deren Lehrpläne sich von denen westlicher Bundesländer meist nicht stärker unterschieden als Letztere untereinander. Auch die im Folgenden beschriebenen Entwicklungen treffen auf die Bundesrepublik und die DDR annähernd gleichermaßen zu; in der DDR beeinflusste die „Neue Mathematik“ jedoch in etwas „abgemilderter“ Weise die Entwicklung des Mathematikunterrichts.

<sup>35</sup>Der Begriff „Aufgabeninseln“ wird hier im Sinne von TIETZE verwendet, der darunter „zusammen-

### 1.2.2 Einbeziehung der Vektorrechnung in den Unterricht

In den fünfziger Jahren des 20. Jahrhunderts wurden in zunehmendem Maße Forderungen erhoben, im Mathematikunterricht Elemente der Vektorrechnung zu behandeln. Damit war (zunächst) der anschauliche und rechnerische Umgang mit „konkreten“ (einem ausgewählten Vektorraummodell, vor allem dem Pfeilklassenmodell, angehörenden) Vektoren sowie mit ihren Produkten intendiert (vgl. [247], S. 98f.). Begründet wurden die Forderungen nach der Einführung der Vektorrechnung vor allem mit folgenden Argumenten (vgl. [81], S. 41f. und [276], S. 53f.):

- Notwendigkeit der Einbeziehung neuerer Inhalte und Vorgehensweisen der Mathematik in den Unterricht sowie der Annäherung des schulischen Mathematikunterrichts an die Anfängervorlesungen an den Hochschulen,
- Bedeutung der Vektorrechnung in der Physik und im Physikunterricht der Schule,
- Eleganz und Einfachheit vektorieller Beweise von Sätzen der Geometrie, die zu Unterrichtsinhalten der Mittelstufe zählen,
- Einfachheit der analytischen Behandlung räumlicher Objekte mithilfe von Vektoren gegenüber ihrer rein koordinatengeometrischen Behandlung;<sup>36</sup>
- Möglichkeit der Vereinheitlichung von ebener und räumlicher Geometrie mithilfe der Vektorrechnung.

Die Mehrzahl der in den fünfziger Jahren erhobenen Forderungen nach Einbeziehung der Vektorrechnung zielte nicht darauf ab, die Koordinatengeometrie vollständig zu ersetzen, sondern darauf, sie durch vektorielle Vorgehensweisen zu ergänzen. Die Vektorrechnung sollte nach den Vorstellungen vieler ihrer Befürworter (wie ATHEN, BAUR und STENDER) nicht als eigenständiges, isoliertes Stoffgebiet gelehrt, sondern zu einem wichtigen Prinzip des Unterrichts bereits in der Mittelstufe werden und durch ihre „*unmittelbare Anschaulichkeit, begriffliche Prägnanz und arbeitstechnische Ökonomie den Unterricht ... vereinfachen, verkürzen und verlebendigen*“ (LOHMEYER 1961, zit. nach [247], S. 99). Dabei wurde sogar daran gedacht, dass vektorielle Methoden in ähnlicher

---

*hängende Komplexe von Aufgaben*“ versteht, „*die sich in ähnlicher Weise lösen lassen*“ ([247], S. 95). Er stellte auf der Grundlage von Schulbuchanalysen sowie verschiedener Ausgaben der „*Methodik des mathematischen Unterrichts*“ (1916–1968) von LIETZMANN fest, „*dass der Schulstoff Analytische Geometrie wie die Analysis eine zunehmende Kanonisierung erfährt, die mit einer gewissen stofflichen Erstarrung einhergeht. An die Stelle eines genetischen Vorgehens tritt damit in zunehmendem Maße eine Art ‚Aufgabendidaktik‘*“ ([247], S. 96). Anhand des Schulbuches Lambacher/Schweizer von 1954 benannte TIETZE die beiden bei der Behandlung der analytischen Geometrie dominierenden Aufgabeninseln: Aufgaben zu Tangenten, Pol und Polare sowie zu geometrischen Örtern. In dem Lehrbuch finden sich ca. 40 Seiten zur ebenen Koordinatengeometrie von Strecken und Geraden sowie 140 Seiten zu den Kegelschnitten. SCHUPP kritisierte das zunehmende „*aufgabenorientierte Durchexerzieren*“ der Kegelschnitte und sah darin eine wesentliche Ursache dafür, dass ein „*derart ergiebiger Themenkreis innerhalb weniger Jahre aus dem Pflichtcurriculum herausgedrängt werden konnte*“ ([234], S. 62).

<sup>36</sup>FREUDENTHAL beschrieb dieses Motiv für die Einführung der Vektorrechnung aufgrund von Erfahrungen in der Hochschullehre folgendermaßen: „*Der Vektorkalkül in bescheidenen Ausmaßen war schon ein großer Fortschritt gegenüber der alten ‚analytischen Geometrie‘ gewesen; der Generation, die in den 20er Jahren ihr Universitätsstudium anfang, war er eine wahre Offenbarung*“ ([84], S. 392).

Weise ein übergreifendes Prinzip des Mathematikunterrichts werden könnten, wie dies bereits seit den Meraner Reformen funktionale Zusammenhänge waren.

In den Lehrplänen der fünfziger Jahre fand sich die Vektorrechnung oft als fakultatives oder wahlobligatorisches Stoffgebiet; obligatorisch war ihre Behandlung zunächst nur in Hamburg und West-Berlin (jeweils in der Klassenstufe 10, vgl. [276], S. 54) sowie bereits ab 1951 im mathematisch-naturwissenschaftlichen Zweig der erweiterten Oberschulen (EOS) in der DDR.<sup>37</sup> In den späten fünfziger und in den sechziger Jahren setzte sich die Vektorrechnung dann als obligatorisches Stoffgebiet in der Sekundarstufe II durch; damit einher ging die Entfernung der sphärischen Trigonometrie und der Kegelschnitte (zunächst ihrer synthetischen Behandlung) aus den Lehrplänen.

In vielen Schulbüchern wurde die aufkommende Vektorrechnung zunächst durch die Hinzunahme weitgehend isolierter Kapitel oder die Herausgabe von Ergänzungsheften berücksichtigt. Eine Durchdringung der analytischen Geometrie mit vektoriellen Methoden erfolgte zunächst nicht, häufig wurde sie sogar doppelt (zuerst als Koordinatengeometrie und dann vektoriell) behandelt (vgl. [276], S. 54f.). Das ursprüngliche Ziel, durch die Einbeziehung der Vektorrechnung den Mathematikunterricht eleganter und effektiver zu gestalten, konnte auf diese Weise natürlich nicht erreicht werden. Vielmehr wuchs der Stoffumfang und die Seitenzahlen der Schulbücher vergrößerten sich trotz komprimierter Darstellung (vgl. [247], S. 99).

Überwiegend verwendeten die Schulbücher für die Einführung von Vektoren das Pfeilklassenmodell und knüpften an die in der Physik für die Darstellung von Kräften genutzten Vektoren an. Geometrische Überlegungen (Betrag und Richtung sowie Richtungssinn) hatten dabei eine tragende Bedeutung. Auch die lineare Abhängigkeit (Kollinearität bzw. Komplanarität) sowie das Skalarprodukt und das Vektorprodukt wurden in den Schulbüchern der fünfziger und sechziger Jahre hauptsächlich geometrisch motiviert und eingeführt (vgl. [81], S. 41; [247], S. 100 und [276], S. 55). Die umfangreichsten „Aufgabeninseln“ bildeten in den Schulbüchern bereits damals Parametergleichungen von Geraden und Ebenen, im Zusammenhang damit Lageuntersuchungen und Schnittberechnungen sowie Normalengleichungen und unter deren Verwendung durchgeführte Abstands- und Winkelberechnungen. Die u. a. bei Schnittberechnungen auftretenden linearen Gleichungssysteme wurden zunächst i. Allg. durch die Übertragung von in der Mittelstufe behandelten Methoden (vor allem Gleichsetzungs- und Einsetzungsverfahren) gelöst; in Einzelfällen kamen Determinanten zur Anwendung (vgl. [247], S. 101).

---

<sup>37</sup>Hier waren in Klasse 10 insgesamt 10 Stunden für die Einführung in die Vektorrechnung bis hin zur Behandlung des Skalarproduktes und des Vektorproduktes vorgesehen. Es stellte sich schnell heraus, dass die in Bezug auf die Relation zwischen Stofffülle und zur Verfügung stehender Zeit unrealistischen DDR-Lehrpläne von 1951 in der Praxis nicht umsetzbar waren, was zu mehreren Korrekturen in den fünfziger Jahren führte. Als ein Ergebnis wurde 1954 die Behandlung der Vektorrechnung von Klasse 10 nach Klasse 11 verlagert. Eine tragfähigere Integration der Vektorrechnung in den Mathematikunterricht der EOS gelang dann 1963, vgl. [81], S. 41.

### 1.2.3 Axiomatisch begründete lineare Algebra im Zuge der „Neuen Mathematik“

In den sechziger Jahren des 20. Jahrhunderts setzten die nach den Meraner Reformen ersten grundlegenden Veränderungen des Mathematikunterrichts in Deutschland und einer Reihe weiterer Länder ein. Zu diesem Zeitpunkt hatten sich die logischen und strukturellen Grundlagen der heutigen Mathematik herausgebildet und wurden durch die Gruppe BOURBAKI in einem strengen logischen Stil und mit einer effizienten Symbolik systematisch zusammengefasst.<sup>38</sup> Axiomatisch begründete Strukturen waren spätestens in den fünfziger Jahren Grundlage der meisten Hochschulvorlesungen in Mathematik geworden. Bei vielen Mathematikern bildete sich die Meinung heraus, dass eine ähnliche Modernisierung auch im Schulunterricht stattfinden müsse und eine sowohl tragfähigere als auch effizientere Gestaltung des Unterrichts ermöglichen könne.

Die IMUK, welche nach dem 2. Weltkrieg ihre Arbeit 1950 unter dem Vorsitz von CHOQUET wieder aufgenommen hatte, schlug 1954 vor, den Mathematikunterricht von der Grundschule an zu reformieren und ihm die Begriffe Menge, Abbildung und Struktur zugrunde zu legen. Einige Jahre später erlangten diese Vorschläge Einfluss auf die Entwicklung der Curricula.<sup>39</sup> Besonders in Frankreich und Belgien entstanden unter dem Einfluss u. a. von DIEUDONNÉ, CHOQUET und PAPY<sup>40</sup> in den sechziger Jahren neue Curricula und Lehrbücher für die Mittel- und Oberstufegeometrie, die von FRANK zusammenfassend folgendermaßen charakterisiert wurden (vgl. [81], S. 41):

- *„Direkter Lehrgegenstand sind die Begriffe Menge, Relation, Abbildung, Gruppe, Vektorraum ...*
- *Geometrie ordnet sich ... dem abstrakten, weitgehend algebraischen Vorgehen unter, wobei der Abstraktionsgrad durch das Formulieren von Axiomen noch erhöht wird. Als unmittelbare Folge davon ist anzusehen:*
  - *eine auffallende Figurenarmut – es dominieren lineare Punktmengen; ...*
  - *viele bekannte und in Anwendungen oft gebrauchte Sätze der ... Geometrie ... finden dadurch wenig Beachtung bzw. werden nicht behandelt.“*

<sup>38</sup>Die Gruppe NICOLAS BOURBAKI (kollektives Pseudonym) führte u. a. die Begriffe injektiv, surjektiv und bijektiv, das Zeichen „ $\Rightarrow$ “ für Implikationen und die bekannten Symbole für die Zahlbereiche ein. 1947 definierte die Gruppe den Begriff des Vektorraumes in der seither verwendeten Form.

<sup>39</sup>1962 stellte sich auf dem Internationalen Mathematikerkongress in Stockholm heraus, dass sich viele Länder für einen Unterricht aussprachen, der Elemente der Mengenlehre und der mathematischen Logik sowie die algebraischen Strukturen Gruppe, Ring, Körper und Vektorraum beinhaltet. Es wurde die Notwendigkeit hervorgehoben, die Sprache im Mathematikunterricht und seine erkenntnistheoretische Struktur zu modernisieren. Kürzungen sollten bei der Elementargeometrie einschließlich der Dreiecksberechnungen, der Trigonometrie und bei den Körperberechnungen erfolgen, vgl. [81], S. 40.

<sup>40</sup>Die Vorschläge der Vertreter der „Neuen Mathematik“ waren keinesfalls identisch. Während DIEUDONNÉ die Geometrie aus den Ideen des Vektorraumes, der quadratischen Form und Gruppen von Abbildungen aufbauen wollte, verfolgten CHOQUET, PAPY und PICKERT den Ansatz, die lineare Algebra auf Grundlage einer geometrischen Axiomatik zu entwickeln (vgl. [245], S. 64ff.). In diesem Bemühen entstanden neue Axiomensysteme der euklidischen Geometrie (vgl. u. a. [65], S. 134ff.). DIEUDONNÉ hielt dieses (u. a. in dem Buch „Neue Elementargeometrie“ von CHOQUET praktizierte) Vorgehen für eine unnötige Konzession an die traditionelle Auffassung von der euklidischen Geometrie ([81], S. 42).



In die Lehrprogramme der Oberstufen höherer allgemeinbildender Schulen wurden fortan fast ausschließlich geometrische Inhalte aufgenommen (bzw. darin belassen), die sich der Vorgehensweise, Geometrie auf der Grundlage einer axiomatisch begründeten linearen Algebra zu betreiben, unterordnen ließen.<sup>41</sup> Allerdings war die Konsequenz, mit der die Algebraisierung des Geometrieunterrichts betrieben wurde, zwischen verschiedenen Ländern sehr unterschiedlich. Während in Frankreich und Belgien zeitweise sogar die Mittelstufengeometrie auf eine algebraisch-axiomatische Grundlage gestellt wurde, erfolgten z. B. in Großbritannien zurückhaltendere Reaktionen auf die Vorgehensweisen der „Neuen Mathematik“. So formulierte FLETCHER 1974 (vgl. [77], S. 21):

*„Die Geometrie macht in der Praxis Gebrauch von einer großen Zahl an Techniken, von denen keine so mächtig ist, dass sie die übrigen überflüssig macht. Zu jedem Zweig der Geometrie gehört eine Auswahl von Problemen, die sich mit Hilfe der jeweiligen Methoden sehr leicht lösen lassen. Wenn man sich umgekehrt auf gegebene Methoden beschränkt, dann kann man leicht Probleme finden, die mit diesen Methoden schwierig zu lösen sind. ... Einige moderne Reformprogramme legen großen Wert auf eine umfassende kohärente Sicht auf die Mathematik. Zweifellos ist dies wünschenswert; aber der Schüler braucht auch Fähigkeiten zur Lösung von Problemen, und er muss auch die allgemeine Stellung der Mathematik in unserer Kultur würdigen können.“*

FLETCHER wandte sich somit gegen die u. a. von CHOQUET vertretene Auffassung, dass die lineare Algebra der „Königsweg“ zur Geometrie sei. Bereits 1973 formulierte der holländische Mathematiker und Didaktiker FREUDENTHAL nach einer ausführlichen Analyse der Einführung geometrischer Begriffe (vor allem des Winkelbegriffs) auf der Grundlage von Vektorräumen und affinen Räumen seine ablehnende Haltung gegenüber einer Unterordnung der Geometrie unter die lineare Algebra (vgl. [84], S. 411):

*„Die Geometrie, die mit linearer Algebra auf der Schule möglich ist, ist ein trübes Abwasser. Der Höhepunkt ist etwa, zu beweisen, dass zwei verschiedene Geraden einen oder keinen Schnittpunkt haben, und dass diese Zahlen für Kreise 0, 1, 2 sind.“*

Die Meinungen zur Behandlung der analytischen Geometrie und linearen Algebra auf der Grundlage eines axiomatisch eingeführten Vektorraumbegriffs waren zu keinem Zeitpunkt einhellig. Dennoch konnte sich dieser Ansatz in den siebziger Jahren in der Bundesrepublik in hohem Maße durchsetzen.<sup>42</sup> TIETZE und WITTMANN dokumentierten diese Entwicklung anhand von Schulbüchern, deren Erstauflagen zwischen 1973 und 1979 erschienen (siehe [247], S. 102ff. und [276], S. 56). Der Aufbau dieser Bücher ist u. a. durch folgende Merkmale gekennzeichnet:

- Der Vektorraumbegriff wird axiomatisch eingeführt und (meist recht knapp) durch Beispiele (Pfeilklassen,  $n$ -Tupel, Vektoren in der Physik) motiviert bzw. illustriert.

<sup>41</sup>Elemente synthetischer Geometrie, die in den nach den Meraner Reformen eingeführten Lehrprogrammen reichhaltig vertreten waren, verschwanden aus den Oberstufencurricula, falls sie nicht bereits zuvor im Zuge der Einführung der Vektorrechnung oder aus Gründen der Zeitknappheit eliminiert wurden. Die Behandlung der Kegelschnitte wurde erheblich eingeschränkt oder gestrichen.

<sup>42</sup>Bereits 1968 verabschiedete die KMK „Empfehlungen und Richtlinien zur Modernisierung des Mathematikunterrichts an den allgemeinbildenden Schulen“, welche u. a. die Behandlung algebraischer Strukturen verlangten. Verbindlich wurden diese Empfehlungen 1972 (siehe [121], S. 239).

- Einen breiten Raum nehmen axiomatisch-deduktive Herleitungen von Sätzen ein.
- Die lineare Abhängigkeit bzw. Unabhängigkeit von Vektoren wird allgemein definiert und hat einen hohen Stellenwert u. a. für Fragen der Lösbarkeit linearer Gleichungssysteme.
- Es erfolgt eine axiomatische Einführung des Skalarproduktes als positiv definite, symmetrische Bilinearform.
- Auch die Theorie der affinen Punkträume wird axiomatisch-deduktiv entwickelt.
- Lineare und affine Abbildungen sind Gegenstand der Schulbücher; bei der Untersuchung von konkreten Beispielen und Eigenschaften erfolgt häufig eine Beschränkung auf den zweidimensionalen Fall. Im Zusammenhang mit den linearen Abbildungen findet die Behandlung von Matrizen und Determinanten statt.

Auch in der DDR erfolgte mit den 1972 in Kraft getretenen Lehrplänen eine axiomatische Einführung des Vektorraumbegriffs im Unterricht der Klasse 12. Allerdings fiel bald auf, dass es sich hierbei um eine „theoretische Überhöhung“ handelte (vgl. [81], S. 62). 1979 erfolgte eine Lehrplankorrektur, nach der Vektoren wieder ausgehend von Verschiebungen als Pfeilklassen eingeführt wurden.<sup>43</sup>

Bereits in der zweiten Hälfte der siebziger Jahre wurde deutlich, dass die axiomatisch-deduktive Behandlung der linearen Algebra die Mehrzahl der Schüler aufgrund ihrer Abstraktheit, die auf nur wenigen konkreten Beispielen gründete, stark überforderte. Für die axiomatische Definition des Vektorraumbegriffs konnte im vorausgehenden Unterricht keine hinreichende Abstraktionsbasis geschaffen werden. TIETZE arbeitete 1979 anhand recht erschreckender Untersuchungsergebnisse bei Studienanfängern des Faches Medizin heraus, dass die „*Transferbreite formaler Sachverhalte, die in der Schule gelernt werden,*“ gering war (vgl. [244]). Die Studierenden konnten zu einem großen Teil recht einfache Aufgaben zu Geradengleichungen in der Ebene nicht lösen. Anhand von Ergebnissen der Transferforschung kam TIETZE zu dem Schluss, „*ein Satz wie ‚Umgang mit formalen Strukturen schule das logische Denken‘ ist schlichtweg falsch.*“

Die DMV äußerte sich in einer Denkschrift zum Mathematikunterricht an Gymnasien aus dem Jahre 1976 besorgt über die „unzulängliche mathematische Vorbildung der meisten Studienanfänger“ und vertrat folgende Auffassung:

„*Die DMV begrüßt es, dass die in den Schulen gelehrt Mathematik an begrifflicher Klarheit gewonnen hat. Jedoch muss das Verständnis der für ein Gebiet typischen Probleme, Grundgedanken und Methoden ... unbedingt Vorrang besitzen vor formaler Exaktheit und Vollständigkeit, die ohnehin im jeweiligen Kenntnisstand der Schüler Grenzen finden und nur schrittweise*

<sup>43</sup>In dem auf dem Lehrplan von 1979 basierenden Schulbuch [82] fällt vor allem die ausführliche Diskussion des Äquivalenzklassenkonzepts bei der Einführung des Vektorbegriffs auf. Im Mittelpunkt stehen dabei Analogiebetrachtungen zwischen Pfeilen und Brüchen sowie zwischen Pfeilklassen und gebrochenen Zahlen. Bei der Behandlung der Vektorrechnung wird das Bemühen um geometrische Veranschaulichungen und Beispiele deutlich; dennoch bleibt auch hier die analytische Geometrie der Vektorrechnung untergeordnet, die auf S. 19 wiedergegebene Äußerung von FREUDENTHAL trifft hinsichtlich der in dem Schulbuch untersuchten geometrischen Objekte in vollem Umfang zu.

erreicht werden können. ... ‚Modernisierung‘ des Mathematikunterrichts darf nicht dahingehend missverstanden werden, dass mathematische Sachverhalte und Begriffe aus der Fachwissenschaft in den Schulstoff übernommen werden, ohne dass ihre Leistungsfähigkeit dort sichtbar wird.“ ([48], S. 71)

Sie forderte eine stärker geometrisch ausgerichtete Behandlung der linearen Algebra / analytischen Geometrie und die Herstellung von Beziehungen zur Analysis:

„Zur Behandlung der (linearen) Gleichungssysteme gehört deren geometrische Interpretation in Ebene und Raum, einerseits um eine anschauliche Vorstellung von der Lösungsmenge zu geben, andererseits weil die analytische Beschreibung der Geometrie des Raumes (nicht nur der Ebene) ein selbständiges Unterrichtsziel ist. Dabei ist die geometrische Interpretation der Vektoren wichtig, die ja auch zu einem besseren Verständnis der Vektorrechnung beiträgt ... . Der Zusammenhang zwischen der Metrik des Raumes und dem Satz des Pythagoras muss selbstverständlich bekannt sein. Bei der Lösung metrischer Aufgaben sollen sowohl die Linearitätseigenschaften des Skalarproduktes wie dessen geometrische Interpretation (Kosinussatz) verfügbar sein.

Schließlich sollte aus der analytischen Geometrie bekannt sein, dass die einfache lineare Abbildung  $(x; y) \mapsto (ax; by)$  z. B. den Einheitskreis auf die Ellipse  $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$  abbildet. Beziehungen zwischen Geometrie und Analysis müssen hergestellt werden können.“ ([48], S. 77)

Von den Ende der siebziger Jahre zunehmenden kritischen Stimmen hinsichtlich der „Neuen Mathematik“ sei noch ZEITLER erwähnt, der auf dem 4. IMUK-Kongress 1980 folgende Meinung äußerte:

„Ich glaube, dass die Eliminierung der Elementargeometrie und der analytischen Geometrie aus unseren Schulen ein gewaltiger Irrtum war. ... Die langweilige, eintönige Berechnung der Schnittelemente von Geraden und Ebenen in verschiedenster Lage ... genügt nicht. ... Ich fordere also weniger Axiomatik, weniger Formalismus, weniger Struktur – dafür mehr Anschauung, mehr Substanz, mehr Anwendung. ... Geometrie muss für die Schüler ein Zaubergarten sein und nicht ein Exerzierplatz.“ ([283], S. 11)

#### 1.2.4 Analytische Geometrie und lineare Algebra im Unterricht nach dem Scheitern der „Neuen Mathematik“

Nachdem spätestens Anfang der achtziger Jahre das Scheitern einer axiomatisch begründeten linearen Algebra im Schulunterricht offensichtlich geworden war, setzten Korrekturen ein. Dabei sind zwei Haupttendenzen zu beobachten:

- Die „Rückkehr“ zu einer vektoriellen analytischen Geometrie bildete die Hauptströmung bei der Umgestaltung der Kurse Lineare Algebra / Analytische Geometrie in den achtziger Jahren und hat die Entwicklung bis heute (2006) am stärksten beeinflusst. Dabei erfolgt zum größten Teil eine vektorielle Behandlung der Raumgeometrie. Stoffgebiete bzw. Kapitel zu einer vektorfreien, „klassischen“ Koordinatengeometrie finden sich in Rahmenplänen (vgl. Abschnitt 1.3) und Schulbüchern relativ selten.<sup>44</sup> Die analytische Geometrie beschränkt sich weit-

<sup>44</sup>Zu den Ausnahmen gehören in dieser Hinsicht die Schulbücher [223] und [260] für Grundkurse

gehend auf die Untersuchung von Geraden und Ebenen sowie ansatzweise von Kreisen und Kugeln.<sup>45</sup> Am Ausgangspunkt der von ZEITLER 1980 geäußerten Kritik (siehe S. 21) sowie der auf S. 19 zitierten Äußerung von FREUDENTHAL haben sich, von Ausnahmen abgesehen, keine wesentlichen Änderungen ergeben.

- Bereits in der zweiten Hälfte der siebziger Jahre gab es Bestrebungen, den Schülern eine *anwendungsorientierte lineare Algebra* zu vermitteln, wobei ein Hauptziel darin bestand, die Schüler durch überzeugende Anwendungen für die Behandlung der linearen Algebra zu motivieren. Dabei wurde vor allem die Behandlung nichtgeometrischer Anwendungen, u. a. aus den Wirtschaftswissenschaften, ins Auge gefasst (vgl. die zusammenfassenden Darstellungen in [245], S. 68ff.; [247], S. 105ff. und [276], S. 59ff.). TÖRNER schlug folgenden Aufbau anwendungsorientierter Kurse in linearer Algebra vor (vgl. [249], S. 21):<sup>46</sup>

„A: Geometrie-unabhängige Einführung des Vektorkalküls innerhalb einer breiten, anwendungsnahen Behandlung der linearen Gleichungssysteme;

B: Ausführliches Arbeiten in und mit dem Matrix-Vektor-Kalkül;

C: Vektorielle Koordinatengeometrie im  $\mathbb{R}^3$ ;

D: Spätes Hinführen zum Vektorraumbegriff.“

Im Zusammenhang mit der Orientierung auf die Vermittlung nichtgeometrischer Anwendungen der linearen Algebra wurde die Frage nach der Art der Einführung von Vektoren und der dazu verwendeten Modelle (geometrische Modelle wie Pfeilklassen oder  $n$ -Tupel) ein wichtiger Gegenstand der Diskussion.<sup>47</sup>

---

sowie [224] für Leistungskurse, die in der zweiten Hälfte der neunziger Jahre erschienen und jeweils ein einführendes Kapitel (25–30 Seiten) mit vektorfreier Koordinatengeometrie enthalten. Allerdings bauen die Bücher im Folgenden kaum auf diese Kapitel auf, da entsprechende Stoffgebiete in den meisten Rahmenplänen nicht vorgesehen sind (vgl. Abschnitt 1.3) und die Kapitel zur vektoriellen analytischen Geometrie auch ohne die Inhalte zur klassischen Koordinatengeometrie nutzbar sein sollen.

<sup>45</sup>Das Grundgerüst der meisten Schulbücher, die von 1980 bis heute erschienen, besteht aus den Kapiteln Vektoren (meist als Pfeilklassen eingeführt); Geraden und Ebenen; Skalarprodukt; Kreise und Kugeln. Diese „Minimalvariante“ findet sich in ähnlicher Form in dem bereits erwähnten DDR-Schulbuch [82] von 1981 und dem bis heute in den meisten Berliner Grundkursen verwendeten Buch von BIGALKE / KÖHLER ([27], 1990). Der Grundkursband [217] der Reihe Lambacher Schweizer (1990) enthält ein ergänzendes Kapitel zu Ellipsen bei einem ansonsten ähnlichen Aufbau. Lineare Gleichungssysteme werden in die Kapitel zu Geraden und Ebenen eingebettet oder separat (oft am Anfang) behandelt. Viele Leistungskursbücher enthalten Kapitel zu affinen Abbildungen. Die Kegelschnitte finden sich in den meisten Lehrwerken für Leistungskurse als ergänzende Kapitel (wie in [52], [97] und [224]) oder neuerdings auf Begleit-CDs wie in [261]. Der Lambacher Schweizer Leistungskurs [216] von 1988 behandelte nur die Ellipse (und zwar als affines Bild des Kreises); in der Ausgabe [8] dieses Lehrwerkes von 2004 tritt auch die Ellipse nicht mehr auf. Die Vektorraumaxiome enthalten viele Bücher für Leistungskurse als rückblickende Zusammenfassungen der Eigenschaften von Vektoren.

<sup>46</sup>Umgesetzt wurde TÖRNERs Konzept in dem 1980 erstmals erschienenen Schulbuch [4]. Dessen zweiter Auflage von 1984 liegt zwar derselbe Aufbau zugrunde, die vektorielle Raumgeometrie erhielt jedoch einen deutlich höheren Stellenwert.

<sup>47</sup>Da diese Frage auch im Hinblick auf die Einbeziehung von Elementen der Computergrafik (bzw. der Informatik allgemein) in das Stoffgebiet Analytische Geometrie / Lineare Algebra von Bedeutung ist, wird darauf in der vorliegenden Arbeit in Abschnitt 3.3.1 noch näher eingegangen.

Bemühungen, eine tragfähige Verbindung von vektorieller analytischer Geometrie des Raumes und anwendungsorientierter linearer Algebra zu finden, sind bereits seit dem Ende der siebziger Jahre zu registrieren.<sup>48</sup> Bei der Neugestaltung der Rahmenpläne wurden die Schwerpunkte jedoch klar zugunsten einer vektoriellen analytischen Geometrie des Raumes gesetzt, nichtgeometrische Anwendungen finden sich auch heute in den meisten Rahmenplänen nur vereinzelt oder in wahlobligatorischen bzw. Ergänzungskursen (siehe Abschnitt 1.3).<sup>49</sup>

Neben der Frage nach einer vektoriell-geometrischen oder anwendungsorientierten Ausrichtung des Unterrichts in analytischer Geometrie / linearer Algebra wurden in den vergangenen 25 Jahren vielfältige weitere Aspekte der Gestaltung dieses Stoffgebietes diskutiert. Da diese Diskussionen nicht als abgeschlossen betrachtet werden können, erscheint der Versuch einer chronologischen Darstellung nicht sinnvoll. Stattdessen werden die für das Thema der vorliegenden Arbeit relevanten Entwicklungen und Vorschläge – nach einer kurzen Darstellung des gegenwärtigen Standes der Curricula – in Abschnitt 1.4 themenbezogen analysiert.

### 1.2.5 Zusammenfassung und einige Schlussfolgerungen

Einhundert Jahre nach dem Erscheinen der ersten Meraner Lehrplanempfehlungen sind die zentralen Entwicklungsrichtungen der Meraner Reformen in hohem Maße aktuell. Auch für den Unterricht in analytischer Geometrie haben die Zielsetzungen

- Entwicklung des Denkens in funktionalen Zusammenhängen;
- Befähigung der Schüler zur Bearbeitung anspruchsvoller Anwendungen und zur Herstellung von Querverbindungen zu anderen mathematischen sowie außermathematischen Inhalten;

---

<sup>48</sup>Derartige Überlegungen wurden u. a. in dem Beitrag [204] von PROFKE angestellt und fanden in dem 1986 erschienenen Schulbuch [97] eine Umsetzung. Das Buch führt Vektoren als Listen ( $n$ -Tupel) ein, wobei Verpackungen mit verschiedenen Anzahlen unterschiedlicher Modellbahnenteile diesen Zugang motivieren. Nach einer recht schnell erfolgenden Einführung des Skalarproduktes (anhand desselben Modells) und einer Behandlung von Rechenregeln werden dreidimensionale Vektoren geometrisch gedeutet. Es folgt ein Kapitel zu linearen Gleichungssystemen und Matrizen, in dem vor allem vielfältige nichtgeometrische Anwendungen (u. a. Stücklisten, Stromkreise und die Interpolation von Funktionen) zum Tragen kommen. Es folgen Kapitel zur vektoriellen analytischen Geometrie, zu affinen Abbildungen und zu Kegelschnitten. Dem Vektorraumbegriff ist ein zweiseitiger Anhang gewidmet.

<sup>49</sup>Auch in den meisten neueren Schulbüchern dominiert klar der Aspekt der vektoriellen Raumgeometrie, vgl. Fußnote 45 auf S. 22. Einige Bücher enthalten Kapitel zu nichtgeometrischen Anwendungen, wie z. B. [223] zur linearen Optimierung, die meist ergänzend am Ende stehen. Die Einführung in die Vektorrechnung wird fast immer geometrisch vorgenommen. Dies gilt auch für eines der derzeit neuesten Schulbücher, die Ausgabe von 2004 des Lambacher Schweizer Leistungskurses ([8]). Zwar wurde der Name „Analytische Geometrie mit Linearer Algebra“ der Ausgabe von 1988 ([216]) in „Lineare Algebra mit analytischer Geometrie“ geändert, inhaltlich spiegelt sich dies jedoch nur in Ansätzen wider. So fanden vereinzelte nichtgeometrische Anwendungsbeispiele Aufnahme in das Buch und es wurde ein abschließendes (24-seitiges) Kapitel „Prozesse und Matrizen“ mit Anwendungsbeispielen u. a. aus der Wirtschaftslehre, der Datenanalyse und der Stochastik hinzugefügt. Hinsichtlich seines prinzipiellen Aufbaus (Vektoren als Pfeilklassen, vektorielle affine und metrische Geometrie des Raumes) folgt jedoch auch dieses Buch dem Ansatz einer vektoriellen Raumgeometrie.

- Förderung des Anschauungs- und Vorstellungsvermögens<sup>50</sup>

nach wie vor eine hohe Bedeutung, werden aber in den gegenwärtigen Curricula und in der Unterrichtspraxis vielfach nicht in ausreichendem Maße berücksichtigt und umgesetzt. Die drei genannten Schwerpunkte werden sich an vielen Stellen der vorliegenden Arbeit wiederfinden; sie sind sowohl für die grundsätzliche Gestaltung des Stoffgebietes Analytische Geometrie als auch für Detailfragen der Behandlung einzelner mathematischer Inhalte relevant.

Das Scheitern der Ideen der „Neuen Mathematik“ ist rückblickend betrachtet m. E. vor allem darauf zurückzuführen, dass im Zeitrafftempo versucht wurde, Entwicklungen der Mathematik auf die Schulmathematik zu übertragen. Die Tatsache, dass algebraische Strukturen eine lange Genese benötigten und sich als Abstraktionen konkreter Beispiele etablieren konnten, fand dabei nicht in ausreichendem Maße Berücksichtigung. Aus dieser Sicht müsste es geradezu verwundern, wenn Schüler der Mittelstufe in der Lage wären, die Tragweite des Gruppenbegriffs zu verstehen und ihn „mit Inhalt auszufüllen“ oder wenn dies Schülern der Oberstufe für den axiomatisch eingeführten Vektorraumbegriff gelänge.<sup>51</sup> Auch Studierende in Anfängervorlesungen haben damit große Probleme, wie entsprechende Untersuchungen belegen (siehe z. B. [74]).

Die Revision der im Zuge der „Neuen Mathematik“ vorgenommenen Reformen erfolgte in vielen Fällen recht inkonsequent. Während strukturell-axiomatische Herangehensweisen weitgehend zurückgedrängt wurden, blieb die seit den späten sechziger bzw. siebziger Jahren bestehende Armut an behandelten geometrischen Objekten durch die weitgehende Beschränkung auf Geraden und Ebenen erhalten. Auch eine nichtgeometrische Ausrichtung des Stoffgebietes auf die Behandlung anspruchsvoller Anwendungen erfolgte nicht in breitem Umfang. Vielfach wird der Unterricht in analytischer Geometrie bis heute durch das Abarbeiten von Kalkülen bei der Lösung von Standardaufgaben zu Schnittpunkten und -geraden, Abständen und Winkeln dominiert.

Mögliche Entwicklungsrichtungen des Unterrichts in analytischer Geometrie werden auf der Grundlage einer Analyse der aktuellen didaktischen Diskussion am Ende dieses Kapitels in Abschnitt 1.6 skizziert.

---

<sup>50</sup>Dieser Schwerpunkt ist m. E. nicht nur hinsichtlich der in den Meraner Empfehlungen hervorgehobenen Förderung des Raumvorstellungsvermögens zu betrachten. Vielmehr ist die Herausbildung anschaulicher und inhaltlicher Vorstellungen mathematischer Gegenstände in einem weiteren Sinne, den wohl auch KLEIN bei seiner auf S. 14 zitierten Äußerung im Auge hatte, anzustreben.

<sup>51</sup>1977 merkte PROFKE im Hinblick auf die Theorie der Vektorräume an: „*Den Reiz einer übergreifenden Theorie und die durch eine solche Theorie ermöglichte Ökonomie bei der Behandlung analoger Probleme erkennt man nur auf dem Hintergrund von Erfahrungen in verschiedenen, aber analogen Bereichen*“ ([203], S. 208 bzw. [204], S. 21). 1978 fügte er hinzu: „*Kann der Schüler ohne ausreichend fundierte und reichhaltige inhaltliche Vorstellungen überhaupt Vorteile der formalen Mathematik erkennen? Formale Mathematik sollte sich auf inhaltliche Vorstellungen stützen können, damit sie nicht in trockenen Formalismus entartet*“ ([204], S. 22).

## 1.3 Überblick über derzeitige Rahmenpläne für die analytische Geometrie bzw. lineare Algebra

Die im Jahre 2005 gültigen Rahmenpläne für die Stoffgebiete Lineare Algebra bzw. Analytische Geometrie der meisten Bundesländer folgen dem bereits in Abschnitt 1.2.4 beschriebenen Aufbau einer vektoriellen analytischen Geometrie des Raumes und enthalten sowohl für Grund- als auch für Leistungskurse folgende Themengebiete:

- Lineare Gleichungssysteme;
- Vektoren (meist als Pfeilklassen eingeführt), Kollinearität und Komplanarität von Vektoren,<sup>52</sup> einfache Anwendungen der Vektorrechnung;
- Affine Geometrie der Geraden und Ebenen (Geraden- und Ebenengleichungen in Parameter- und Koordinatenform, Untersuchung von Lagebeziehungen, Bestimmung von Schnittpunkten bzw. -geraden);
- Metrische Geometrie der Geraden und Ebenen (Skalarprodukt, Normalenvektoren und -gleichungen, Abstands- und Winkelberechnungen).<sup>53</sup>

Dieser grundsätzliche Aufbau findet sich in den Rahmenplänen für Baden-Württemberg,<sup>54</sup> Bayern, Berlin, Brandenburg, Hessen, Mecklenburg-Vorpommern, Niedersachsen, Nordrhein-Westfalen, Sachsen, Sachsen-Anhalt, Schleswig-Holstein und Thüringen. In Bremen und Rheinland-Pfalz bestehen Wahlmöglichkeiten zwischen einer vektoriellen Raumgeometrie und der vertiefenden Behandlung von Matrizen und Abbildungen sowie nichtgeometrischen Anwendungen von Matrizen und linearen Gleichungssystemen.

In den Rahmenplänen einiger Bundesländer wie Hamburg, Mecklenburg-Vorpommern, Sachsen und Berlin geht der Einführung von Vektoren ein Stoffgebiet zur Koordinatengeometrie der Ebene und teilweise des Raumes voraus. Dabei werden neben Geraden oft auch Kreise in der Ebene behandelt. Die Rahmenpläne der meisten Länder, in denen sich der Aufbau der Stoffgebiete zur analytischen Geometrie bzw. linearen Algebra an einer vektoriellen Raumgeometrie ausrichtet, sehen vor allem für Leistungskurse (LK) die Behandlung von Themenbereichen vor, die über die oben genannten Kerninhalte hinausgehen. Die folgende Aufzählung soll darüber einen groben Überblick vermitteln.

---

<sup>52</sup>Linearkombinationen sowie die lineare Abhängigkeit und Unabhängigkeit von Vektoren, teilweise auch Basen von Vektorräumen, werden in einigen Bundesländern (hauptsächlich in Leistungskursen) explizit thematisiert. Meist geht dies mit einer axiomatischen Definition des Vektorraumbegriffs einher, die allerdings der Behandlung eines Modells (meist des Pfeilklassenmodells) folgt. Die meisten Bundesländer verzichten mittlerweile auf diese strukturellen Betrachtungen, stattdessen werden die Kollinearität und Komplanarität von Vektoren geometrisch und rechnerisch behandelt.

<sup>53</sup>Auf der Internetseite <http://www.afiller.de/habil> zu der vorliegenden Arbeit (siehe Anhang D, S. 365) befinden sich die für den Überblick ausgewerteten, im Jahre 2005 gültigen Rahmenpläne der gymnasialen Oberstufen der meisten Bundesländer (mit Ausnahme derjenigen Länder, für die bislang keine Rahmenpläne in elektronischer Form veröffentlicht wurden). Die jeweils aktuellen Versionen lassen sich über den Deutschen Bildungsserver (<http://www.bildungsserver.de>) auffinden.

<sup>54</sup>Diese Aussage bezieht sich auf den 2005 für die Oberstufe noch gültigen Bildungsplan für Baden-Württemberg, der in den nächsten Jahren klassenstufenweise abgelöst wird, siehe S. 30.

- *Kreise und Kugeln*: Berlin, Hessen (fakultativ), Mecklenburg-Vorpommern (LK), Sachsen (LK), Sachsen-Anhalt, Schleswig-Holstein
- *Vektor- und teilweise Spatprodukt*: Brandenburg (LK), Bayern (LK), Sachsen (LK), Hessen (LK - wahlobligatorisch)
- *Kegelschnitte*: Bayern (Wahlpflichtthema in Klassenstufe 10), Brandenburg (GK und LK), Hamburg, Hessen (Ergänzungsthema)
- *Matrizen und/oder affine Abbildungen*: Brandenburg (LK), Hessen (LK - wahlobligatorisch), Nordrhein-Westfalen (Ergänzungsthema)

Als Beispiel eines Rahmenplanes, der in vielerlei Hinsicht exemplarisch für eine Reihe von Bundesländern steht, jedoch überdurchschnittlich detailliert Lernziele und -inhalte aufschlüsselt, wird im Folgenden der 2005 noch gültige **Berliner Rahmenplan für Grundkurse** betrachtet.<sup>55</sup> Die analytische Geometrie (und die damit eng zusammenhängenden linearen Gleichungssysteme) werden nach dessen Vorgaben an drei Stellen, nämlich am Ende von Klasse 11, am Ende des 2. Kurshalbjahres und im 3. Kurshalbjahr behandelt.<sup>56</sup>

Klasse 11: Einführung in die analytische Geometrie (30 Stunden)	
<b>Hinweis zum Lernabschnitt:</b> Der Lernabschnitt dient einer elementaren Einführung in die analytische Geometrie, vor allem der des Raumes. Vektoren sind naiv über das Koordinatensystem als Spalten einzuführen; an eine Einführung des Vektorbegriffs über Pfeilklassenbildung und an strukturelle Betrachtungen zum Vektorraum ist nicht gedacht.	
Lernziele	Punkte im Koordinatensystem
Wissen, dass man im dreidimensionalen kartesischen Koordinatensystem Punkte durch Zahlentripel beschreiben kann. Die zu vorgegebenen Zahlentripeln gehörenden Punkte im dreidim. kart. KS zeichnen können.	Darstellung von Punkten im $\mathbb{R}^3$ .
Die Entfernung von je zwei Punkten im Koordinatensystem berechnen können.	Entfernungen von Punkten ( $\mathbb{R}^2$ und $\mathbb{R}^3$ ) im kartesischen Koordinatensystem.
Die Gleichungen für Kreis und Kugel in Ursprungslage kennen und herleiten können. Rechnerisch feststellen können, ob Punkte innerhalb oder außerhalb vorgegebener Kreise (Kugeln) bzw. auf ihnen liegen.	Gleichungen von Kreisen und Kugeln mit dem Ursprung als Mittelpunkt.

<sup>55</sup>Beginnend mit dem Schuljahr 2005/06 werden in Berlin für die gymnasiale Oberstufe neue Curricula eingeführt, vgl. S. 29. Die Auswahl des Berliner Rahmenplanes erfolgte u. a. deshalb, weil er die Rahmenbedingungen eines Unterrichtsversuches zur Einbeziehung von Elementen der 3D-Computergrafik in das Stoffgebiet Analytische Geometrie festlegte, der in Abschnitt 5.1 der vorliegenden Arbeit beschrieben wird. Ein weiterer Grund für die etwas ausführlichere Betrachtung der Pläne von Berlin, Baden-Württemberg und Hamburg besteht darin, dass im Zeitraum der Fertigstellung dieser Arbeit die Einführung neuer Curricula in diesen Bundesländern erfolgt und somit der Frage nach gegenwärtigen Tendenzen der Lehrplanentwicklung nachgegangen werden kann.

<sup>56</sup>Unterschiede zwischen dem Berliner Rahmenplan und den Plänen anderer Bundesländer bestehen vor allem in der dreigeteilten Behandlung der analytischen Geometrie/linearen Algebra und in der Einführung von Vektoren als  $n$ -Tupel (der jedoch recht schnell die Pfeildarstellung folgt).



Lernziele	Vektoren im kartesischen Koordinatensystem	Lerninhalte
Wissen, dass Spalten aus zwei oder drei reellen Zahlen Vektoren heißen. Im kartesischen Koordinatensystem Spalten (Vektoren) durch Pfeile deuten können, Pfeile durch Spalten beschreiben können und den Zusammenhang zwischen Punkten und Pfeilen bzw. Spalten kennen.		Vektoren im $\mathbb{R}^2$ und $\mathbb{R}^3$ als Spalten; Zusammenhang zwischen Spalten und Punkten, Pfeilen und Verschiebungen in der Ebene und im Raum.
Mit Vektoren sicher rechnen können.		Rechnen mit Vektoren: Gegenvektor, Addition und Subtraktion von Vektoren, Nullvektor, Multiplikation eines Vektors mit einer Zahl, Länge eines Vektors, Einheitsvektor.
Einige Regeln für das Rechnen mit Vektoren durch Rechnen mit Spalten herleiten können.		Regeln für das Rechnen mit Vektoren.
Anwendungen von Vektoren kennen.		Vektoren in geometrischen und nichtgeometrischen Bezügen.
<b>Geraden</b>		
Den Mittelpunkt einer Strecke vektoriell berechnen können.		Mittelpunkt einer Strecke.
Geradengleichungen in vektorieller Form bestimmen können.		Geradengleichungen (vektoriell) im $\mathbb{R}^3$ : Punkt-Richtungs-Form, Zwei-Punkte-Form. Parallele Geraden.
Rechnerisch überprüfen können, ob vorgegebene Punkte auf einer Geraden liegen.		
Die relative Lage von zwei Geraden zueinander untersuchen und den Schnittpunkt gegebenenfalls berechnen können.		Relative Lage von Geraden zueinander im $\mathbb{R}^3$ : sich schneidende, parallele, windschiefe, identische Geraden.
Einfache geometrische Probleme mit Vektoren lösen können.		Anwendungen von Geradengleichungen zur Lösung geometrischer Probleme, z. B. Schwerpunktbestimmung im Dreieck, Spurpunkte.
Geradengl. im $\mathbb{R}^2$ in vektorieller und in Koordinatenform wechselweise umformen können.		Verschiedene Gleichungen zu einer Geraden im $\mathbb{R}^2$ .
Rechnerisch feststellen können, ob vorgegebene Kreise und Geraden Schnittpunkte haben, und diese gegebenenfalls bestimmen können.		Schnitt von Kreis in Ursprungslage mit einer Geraden im $\mathbb{R}^2$ .

**Grundkurs ma-2: Lineare Gleichungssysteme****(15 Stunden)**

**Hinweise zum Lernabschnitt:** In diesem Abschnitt sollen die Schüler lernen, in Verbindung mit Anwendungen lineare Gleichungssysteme mit  $m$  Gleichungen und  $n$  Variablen sicher zu lösen. Durch abwechslungsreiche Wahl von  $m$  und  $n$  soll die Bearbeitung einer Vielfalt von linearen Gleichungssystemen erreicht werden. Die Begriffe „Lineare Abhängigkeit“ und „Lineare Unabhängigkeit“ stehen nicht zur Verfügung. Der Gebrauch von Determinanten ist nicht vorgesehen.

Lineare Gleichungssysteme mit $m$ Gleichungen und $n$ Variablen (auch $m \neq n, m > 3, n > 3$ ) kennen und wissen, dass Lösungen $n$ -Tupel sind.	Hinführung zu linearen Gleichungssystemen mit $m$ Gleichungen und $n$ Variablen über Anwendungsprobleme (Beispiele: siehe unten).
Den Gauß-Algorithmus kennen und lineare Gleichungssysteme mit seiner Hilfe lösen können.	Gauß-Algorithmus für LGS mit $m$ Gleichungen und $n$ Variablen, auch für $m \neq n$ ; Elementarumformungen unter dem Aspekt, dass sich die Lösungsmenge des LGS nicht ändert.

Lernziele	Lerninhalte
Die Lösungsmenge eines linearen Gleichungssystems angeben können.	Angabe der Lösungsmenge als Menge von $n$ -Tupeln, dabei Berücksichtigung der drei Hauptfälle von leerer, einelementiger und mehrelementiger Lösungsmenge.
Anwendungsaufgaben mit Hilfe von linearen Gleichungssystemen mathematisieren und lösen können.	Anwendungen: Verkehrsnetze, Materialverbrauch, Bestimmung von Funktionen zu vorgegebenen Bedingungen, Mischungsprobleme, Schnittmengen von Geraden im $\mathbb{R}^3$ .

**Grundkurs ma-3: Analytische Geometrie****(30 Stunden)**

**Hinweise zum Lernabschnitt:** Schwerpunkt dieses Lernabschnitts sind das Skalarprodukt und seine Anwendungen. Die Lerninhalte der drei Blöcke A, B, C sind im Unterricht sinnvoll miteinander zu verbinden, also nicht nacheinander zu unterrichten. Die Einführung von Begriffen wie „Lineare Abhängigkeit“ bzw. „Lineare Unabhängigkeit“ ist nicht vorgesehen. An geeigneten Stellen ist das Lösen linearer Gleichungssysteme wieder aufzugreifen; dabei soll eine geometrische Deutung der linearen Gleichungssysteme in drei Variablen und ihrer Lösungsmengen erfolgen. Die Schulung des räumlichen Anschauungsvermögens – auch durch angemessene zeichnerische Darstellung geometrischer Objekte – ist in den Unterricht einzubeziehen.

<b>A.</b> Eine Definition für das Skalarprodukt zweier Vektoren im $\mathbb{R}^3$ kennen.  Regeln für das Rechnen mit dem Skalarprodukt kennen.  Längen und Winkelgrößen mit Hilfe des Skalarproduktes berechnen können.	Skalarprodukt in den Formen $\begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \cdot \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = a_1 \cdot b_1 + a_2 \cdot b_2 + a_3 \cdot b_3$ und $\vec{a} \cdot \vec{b} =  \vec{a}  \cdot  \vec{b}  \cdot \cos(\vec{a}, \vec{b})$ ; Rechenregeln für SP. Berechnung von Längen und Winkelgrößen mit Hilfe des Skalarproduktes; Orthogonalität.
<b>B.</b> Geraden- und Ebenengleichungen im $\mathbb{R}^3$ in Parameterform bestimmen können. Ebenengleichungen in Normalen- bzw. Koordinatenform bestimmen können. Verschiedene Formen von Ebenengleichungen wechselseitig umformen können. Den Abstand einer Ebene vom Ursprung mit Hilfe der Hesseschen NF bestimmen können. Kugelgleichungen bestimmen können.	Geradengleichung in Parameterform im $\mathbb{R}^3$ , Ebenengleichung in Parameterform im $\mathbb{R}^3$ . Normalen- bzw. Koordinatenform der Ebenengleichung. Wechselseitige Umformungen der Parameterform in Normalen- bzw. Koordinatenform der Ebenengleichungen. Hessesche Normalenform der Ebenengleichung.  Gleichung der Kugel in allgemeiner Lage.
<b>C.</b> Rechnerisch überprüfen können, ob vorgegebene Punkte auf einer Geraden, Ebene bzw. Kugel liegen. Schnittpunktmengen bestimmen und geometrisch interpretieren können: Gerade - Gerade, Gerade - Ebene, Ebene - Ebene. Schnittwinkel berechnen können. Abstände berechnen können. Mit Hilfe von Abstandsberechnungen entscheiden, ob eine Ebene eine Kugel in einem Kreis schneidet, berührt oder meidet.	Lagebeziehungen geometrischer Objekte im $\mathbb{R}^3$ , insbesondere Bestimmung von Schnittpunktmengen, Schnittwinkeln und Abständen. Vermischte Aufgaben komplexerer Art. Hinweis: Bei der Schnittpunktmengenbestimmung Ebene - Ebene sollte wenigstens eine Gleichung in Normalen- bzw. Koordinatenform vorliegen.

Häufig stehen die für die analytische Geometrie geplanten 30 Unterrichtsstunden am Ende von Klasse 11 sowie die für lineare Gleichungssysteme vorgesehenen 15 Stun-

den am Ende des Kurses ma-2 aufgrund von Verzögerungen bei der Behandlung der vorangegangenen Stoffgebiete nicht zur Verfügung. Entsprechend lückenhaft sind die Kenntnisse, über die viele Schüler zu Beginn des 3. Kurshalbjahres verfügen.<sup>57</sup>

Für *Leistungskurse* sind nach dem 2005 noch gültigen Berliner Rahmenplan dieselben Themengebiete wie für Grundkurse – jedoch mit erheblichen Ergänzungen – vorgesehen. Für das Stoffgebiet Lineare Gleichungssysteme sind 25 Stunden zu Beginn des 3. Kurshalbjahres eingeplant. Über die für Grundkurse aufgeführten Themen hinaus werden strukturelle Betrachtungen u. a. zum Zusammenhang der Lösungsmengen von inhomogenen und zugehörigen homogenen LGS sowie zum Rang von LGS gefordert. Im Zusammenhang damit steht die Behandlung des Vektorraumbegriffs. Das sich an die Behandlung der LGS anschließende Stoffgebiet Analytische Geometrie (35 Stunden) enthält zusätzlich zu den Inhalten des Grundkurses u. a. die Behandlung des Vektorproduktes. Es folgt ein 15-stündiges Wahlgebiet, für das zwei Alternativen möglich sind:

- *Matrizen* (mit geometrischen und nichtgeometrischen Anwendungen: lineare und affine Abbildungen, Input-Output-Analyse, Stücklistenproblem);
- *Kegelschnitte* (u. a.: Schnitte eines Kegels mit einer Ebene; Brennpunkte, Abstandseigenschaften und Gleichungen von Ellipse, Hyperbel, Parabel).

Beginnend mit dem Schuljahr 2005/06 treten in Berlin schrittweise neue „Curriculare Vorgaben für die gymnasiale Oberstufe“ in Kraft, die für die Behandlung der analytischen Geometrie wesentliche Veränderungen mit sich bringen. So wird im 1. Halbjahr des 11. Schuljahres ein neu konzipiertes Stoffgebiet „*Koordinatengeometrie und Funktionen*“ behandelt, wobei folgende Inhaltsbereiche vorgesehen sind:

- *Koordinatengeometrie der Ebene und des Raumes* (u. a. Abstände von Punkten, Beschreibung von Strecken, Teilungspunkte von Strecken);
- *Geraden im ebenen Koordinatensystem*;
- *Parabeln im ebenen Koordinatensystem*;
- *Kreise im ebenen Koordinatensystem*;
- *Trigonometrische Funktionen*.

Dieses Curriculum folgt somit dem Ansatz einer „klassischen Koordinatengeometrie“ (vgl. Abschnitt 1.1.1) und stellt eine Verbindung zwischen analytischer Geometrie und Funktionenlehre her.<sup>58</sup> Wesentlich weniger gravierend sind die Veränderungen des Stoffgebietes zur vektoriellen analytischen Geometrie in Grundkursen; es finden sich hier hauptsächlich die bereits genannten Inhalte des alten Rahmenplanes.

---

<sup>57</sup>Auf Ergebnisse einer kurzen Überprüfung der Vorkenntnisse von Schülern zu Beginn eines Kurses ma-3 wird bei der Beschreibung eines Schulversuches in Abschnitt 5.1.1 eingegangen.

<sup>58</sup>M. E. sind durch diesen Ansatz gute Chancen gegeben, einige gravierende Defizite des Unterrichts in analytischer Geometrie zu beheben oder zumindest abzumildern (u. a. die Reduktion auf das Lösen von Routineaufgaben und die Formenarmut, siehe 1.4). Auch aufgrund der vorgesehenen Stundenzahl und der Stellung im 1. Halbjahr (wodurch Stundenausfälle unwahrscheinlicher werden) könnte das neue Curriculum für die Behandlung der analytischen Geometrie bessere Perspektiven schaffen.

Für Leistungskurse bestehen, nach einer obligatorischen Einführung von Vektoren und der Behandlung linearer Gleichungssysteme, zwei Wahlmöglichkeiten:

- A1: *Vektorielle analytische Geometrie* (ähnlich dem bereits skizzierten Aufbau);  
 A2: *Anwendung von Matrizen bei Abbildungen* (analytische Beschreibung affiner Abbildungen sowie eine nichtgeometrische Anwendung von Matrizen).

Einem gänzlich anderen Ansatz folgen die 2004 eingeführten und sukzessive für die einzelnen Klassenstufen in Kraft tretenden **Bildungsstandards für Baden-Württemberg**. Diese sind nach Leitideen gegliedert und machen recht weit gefasste Vorgaben zu den anzustrebenden Kompetenzen und zu den Unterrichtsinhalten. Sie enthalten folgende Aussagen, die sich auf die analytische Geometrie beziehen lassen :<sup>59</sup>

#### Klassenstufe 10

*Leitidee „Raum und Form“ (Auszug)*

Kompetenzen: • geometrische Objekte im Raum analytisch beschreiben und ihre Lagebeziehungen analysieren.

Inhalte: – zentrische Streckung, Strahlensätze, Satz des Pythagoras  
 – Berechnung von Streckenlängen und Inhalten bei Körpern  
 – Ortsvektor, Geradengleichung

#### Kursstufe (Klassenstufen 11 und 12)

*Leitidee „Algorithmus“ (Auszug)*

Kompetenzen: • lineare Gleichungssysteme auf Lösbarkeit untersuchen; die Lösungsmenge eines linearen Gleichungssystems bestimmen.

Inhalte: – Gauß-Algorithmus

*Leitidee „Raum und Form“*

Kompetenzen: • geometrische Objekte im Raum vektoriell beziehungsweise analytisch beschreiben und ihre Lagebeziehungen analysieren;  
 • Eigenschaften von geometrischen Objekten und Beziehungen zwischen geometrischen Objekten beschreiben und berechnen.

Inhalte: – Ebenen, Winkel, Abstände

*Leitidee „Messen“ (Auszug)*

Kompetenzen: • das Konzept der Rekonstruktion auf verschiedene Anwendungsfelder übertragen

Inhalte: – Inhalt krummlinig begrenzter Flächen (auch Kreis), Volumen (auch Pyramide, Kegel)

*Leitidee „Vernetzung“*

Kompetenzen: • heuristische Verfahren zur Erkenntnisgewinnung kennen und einsetzen;  
 • mithilfe von Vektoren beweisen.  
 • Probleme lösen, die den Einsatz von Begriffen und Verfahren aus verschiedenen Teilbereichen der Mathematik erfordern.

Inhalte: – Verbindungen zwischen den Teilgebieten Analysis, Geometrie und Stochastik

*Leitidee „Modellieren“ (Auszug)*

Kompetenzen: • inner- und außermathematische Sachverhalte und ihre Veränderungen auch in komplexeren Zusammenhängen mathematisch modellieren.

Inhalte: – Wahl geeigneter Grundobjekte (z. B. Koordinatensystem, Variable)  
 – Anwendungen linearer Gleichungssysteme

<sup>59</sup>Zu den neuen Bildungsstandards von Baden-Württemberg ist anzumerken, dass deren Einführung mit der Verkürzung der Schulzeit auf 12 Jahre einhergeht. Eine Differenzierung in Grund- und Leistungskurse besteht in diesem Bundesland in Mathematik seit dem Abiturjahrgang 2004 nicht mehr.

Die Aussagen zur Leitidee „Raum und Form“ lassen erwarten, dass die Behandlung einer vektoriellen analytischen Geometrie, ergänzt um Anwendungen linearer Gleichungssysteme, auch nach der Einführung der neuen Bildungsstandards einen Schwerpunkt des Unterrichts in Baden-Württemberg bilden wird. Die Offenheit und die wenig detaillierten Vorgaben der Bildungsstandards eröffnen Entscheidungsspielräume für Lehrer, Schwerpunkte des Unterrichts zu bestimmen sowie auch neue Themenkreise und Hergangeweisen einzubringen.<sup>60</sup> Dies bedeutet aber noch nicht, dass diese Spielräume und Chancen auch genutzt werden. Eine wichtige Bedeutung in diesem Zusammenhang wird die zukünftige Gestaltung des Zentralabiturs haben. Vielfach haben die erfahrungsgemäß oft vorkommenden Aufgabentypen im Zentralabitur (zusammengestellt in Sammlungen wie [5]) eine hohe Bedeutung als „heimliche Lehrpläne“ und im Unterricht dominieren häufig Übungen für das Lösen von Standardaufgaben (vgl. Abschnitt 1.4.2). Daran dürften sich durch neue Curricula kaum nennenswerte Veränderungen vollziehen, wenn nicht gleichzeitig eine „Öffnung“ der Abituraufgaben im Sinne der Reduktion des Anteils von Routineaufgaben und der Einbeziehung von Aspekten der Vernetzung sowie des Modellierens und Problemlösens erfolgt.

Abschließend sei noch kurz auf den 2004 verabschiedeten **Hamburger Rahmenplan** eingegangen, der sukzessive bis 2007 in Kraft tritt. Er enthält für die Klassenstufe 11 einen Themenbereich „Geometrie“, für den die Behandlung von Elementen der Koordinatengeometrie der Ebene mit empfohlener Nutzung von dynamischer Geometriesoftware und Funktionenplottern bzw. CAS vorgesehen ist. Gegenstände dieses Stoffgebietes sind u. a. *Kreis, Ellipse, Parabel* und *Hyperbel* sowie *Ausblicke auf Kugel und Paraboloid*. Schwerpunktmäßig hervorgehoben werden *Durchführung und Vergleich von geometrischen und algebraischen Problemlösungen*.

Für die Klassen 12 und 13 sieht der Hamburger Plan einen modularen Aufbau aus wählbaren Themenbereichen vor. Dabei ist sowohl für Grund- als auch für Leistungskurse die Behandlung der *vektoriellen analytischen Geometrie des Raumes* oder eines Stoffgebietes „*Matrizen und Vektoren als Datenspeicher*“ möglich, in dem arithmetische Aspekte von Vektoren und Matrizen sowie Anwendungen im Vordergrund stehen. Die Schwerpunkte dieses Stoffgebietes sind nicht so eng an die Informatik angelehnt, wie es der Name vermuten lässt; die Behandlung rekursiver Verfahren ist jedoch geplant.

Zusammenfassend ist festzustellen, dass sowohl in den neuen Berliner und Hamburger Rahmenplänen für die Klassenstufe 11 als auch in den Bildungsstandards für Baden-Württemberg Tendenzen zur Herstellung von Vernetzungen zwischen der Analysis bzw. den elementaren Funktionen und der analytischen Geometrie bestehen. Auf die Frage, ob sich der in B/W verfolgte sehr offene Ansatz oder die recht konkreten Vorgaben in Berlin als wirksamer erweisen, sind erst in einigen Jahren, wenn die Curricula eingeführt wurden und erste Erfahrungen vorliegen, Antworten zu erwarten.

---

<sup>60</sup>Insbesondere für die in der vorliegenden Arbeit unterbreiteten Vorschläge zur Einbeziehung von Elementen der Computergrafik und entsprechende Schwerpunktverschiebungen des Unterrichts in analytischer Geometrie sind nach den neuen Bildungsstandards des Landes Baden-Württemberg größere Freiräume vorhanden als z. B. auf der Grundlage der „alten“ Berliner Rahmenpläne.

## 1.4 Defizite des Unterrichts in analytischer Geometrie und Ansätze zu seiner Weiterentwicklung

Nachdem in den vorigen Abschnitten die historische Entwicklung und der gegenwärtige curriculare Stand des Unterrichts in analytischer Geometrie / linearer Algebra skizziert wurden, sollen einige Tendenzen seiner Weiterentwicklung anhand in den letzten Jahrzehnten herausgebildeter und nunmehr weitgehend konsensfähiger Grundpositionen zur Stellung und zu den Aufgaben des Mathematikunterrichts herausgearbeitet werden.<sup>61</sup>

### 1.4.1 Analytische Geometrie und Allgemeinbildung

Die Diskussion der letzten Jahre über die Weiterentwicklung des Mathematikunterrichts in der Sekundarstufe II ist von dem Ansatz geprägt, dessen allgemeinbildenden Charakter zu stärken. Dieser ist nach WINTER dadurch gekennzeichnet, dass der Mathematikunterricht drei *Grunderfahrungen* ermöglicht (vgl. [274], S. 37):

- (1) „*Erscheinungen der Welt um uns, die uns alle angehen oder angehen sollten, aus Natur, Gesellschaft und Kultur, in einer spezifischen Art wahrzunehmen und zu verstehen,*
- (2) *mathematische Gegenstände und Sachverhalte, repräsentiert in Sprache, Symbolen, Bildern und Formen, als geistige Schöpfungen, als eine deduktiv geordnete Welt eigener Art kennen zu lernen und zu begreifen,*
- (3) *in der Auseinandersetzung mit Aufgaben Problemlösefähigkeiten, die über die Mathematik hinaus gehen, (heuristische Fähigkeiten) zu erwerben.“*

WINTER forderte, mathematische Inhalte nicht nur nach innerfachlichen Ordnungsprinzipien zu strukturieren, „*sondern auch aus anderen pädagogisch relevanten Blickwinkeln zu sehen und zu verstehen, vor allem aus der Sicht*

- *der historischen Genese von Ideen,*
- *der möglichen Bezüge zu unterschiedlichen außermathematischen Bereichen,*
- *der Akzentuierung nach übergeordneten fundamentalen Ideen,*
- *der möglichen Verwurzelungen in Alltagserfahrungen,*
- *der möglichen unterschiedlichen Repräsentationsformen,*
- *der möglichen Distanzen zu Primärintuitionen und damit zu möglichen Verständnishürden,*
- *der möglichen Erschließbarkeit durch selbständige Lernaktivitäten.“* ([274], S. 46).

Die Forderung, die o. g. drei Grunderfahrungen gleichberechtigt im Mathematikunterricht der Sekundarstufe II zu ermöglichen, hat breite Akzeptanz gefunden und ist explizit in den 2002 von der Kultusministerkonferenz beschlossenen einheitlichen Prüfungsanforderungen im Fach Mathematik (EPA) enthalten (siehe [136]). Hiernach soll

<sup>61</sup>Auf Fragen des Computereinsatzes wird zunächst nur kurz im Zusammenhang mit inhaltlichen Aspekten eingegangen, zu denen sie in unmittelbarer Beziehung stehen. Ausführlicher werden Fragen der Nutzung des Computers im Mathematikunterricht in Abschnitt 1.5 erörtert.

die Berücksichtigung aller drei Grunderfahrungen in jedem der Sachgebiete Analysis, Analytische Geometrie und Stochastik erfolgen. Der allgemeinbildende Wert der analytischen Geometrie wird vor allem in „ihren mächtigen Methoden und interessanten Objekten zur Erschließung des uns umgebenden Raumes“ gesehen.

In Bezug auf die gegenwärtige Situation konstatierten BORNELEIT, DANCKWERTS, HENN und WEIGAND in einer 2001 vorgelegten Expertise zum Mathematikunterricht in der gymnasialen Oberstufe, auf die sich die KMK bei der Erarbeitung der EPA 2002 wesentlich stützte, die „*einseitige (zumindest implizite) Orientierung an der Grunderfahrung (2)*“ und die „*systematische Vernachlässigung von (1) und (3)*“. Als weitere gravierende Defizite des Mathematikunterrichts in der S II wurden die „*einseitige Orientierung am Kalkülaspekt der Mathematik*“ sowie eine „*Unterrichtskultur, die eine aktive Sinnkonstruktion mathematischer Inhalte durch die Lernenden nicht ausreichend unterstützt*“, genannt (vgl. [29], S. 78ff.).

Als inhaltliche Maßnahmen, um den allgemeinbildenden Charakter des Mathematikunterrichts in der S II stärker auszuprägen, schlugen die Autoren vor (vgl. [29], S. 81ff.):

- *Orientierung an fundamentalen Ideen.* Als fundamentale Ideen wurden u. a. räumliches Strukturieren und Koordinatisieren aufgeführt.<sup>62</sup> Die Autoren forderten, „*die kalkülorientierten Teile in Zeitaufwand und Wertigkeit zu Gunsten der inhaltlich orientierten Teile zu reduzieren*“.
- *Vernetzung als Orientierungsgrundlage.* Die Lernenden sollten sowohl *vertikale Vernetzungen* (zwischen der Behandlung derselben Themenbereiche auf verschiedenen Stufen mit unterschiedlichem Abstraktionsgrad und sich verändernden Methoden) als auch *horizontale Vernetzungen* (zwischen den einzelnen Teilgebieten des Unterrichts) erkennen.<sup>63</sup>
- *Grundvorstellungen versus Kalkülorientierung.* Die Autoren forderten eine „*deutliche Trennung zwischen der Idee und Bedeutung eines mathematischen Begriffs oder entsprechender Grundvorstellungen über mathematische Verfahren einerseits und dem kalkülhaften Umgang damit andererseits*“.<sup>64</sup>
- *Anwendungsorientierung.* Anwendungen der Mathematik sollten realitätsadäquat in den Unterricht einbezogen werden, wobei die Schüler den Modellbildungskreislauf bewusst durchlaufen und an konkreten Beispielen erleben.<sup>65</sup>

<sup>62</sup>Zu fundamentalen Ideen der linearen Algebra und analytischen Geometrie siehe 1.4.3.

<sup>63</sup>Die Autoren führten folgendes Beispiel an: „*Auf eine isolierte Axiomatisierung des Vektorraums mit der Reduzierung auf lineare geometrische Gebilde sollte zugunsten einer an die Geometrie der S I anknüpfenden inhaltlich orientierten analytischen Geometrie des uns umgebenden Raums verzichtet werden. Auch hier sollte viel stärker die Vernetzung von Analysis und Analytischer Geometrie hervortreten, indem etwa Kurven und insbesondere die Kegelschnitte wieder zentrale Objekte des Mathematikunterrichts werden.*“ Sie verwiesen dazu auf Forderungen und Vorschläge von SCHUPP ([234]).

<sup>64</sup>Als Beispiel nannten die Autoren u. a. die Darstellung geometrischer Gebilde (Geraden, Ebenen, Kreise, Ellipsen,...) mithilfe analytischer Methoden und das formale Lösen von Gleichungssystemen.

<sup>65</sup>Auf einige Vorschläge zur Modellierung von Sachsituationen und zu Anwendungen im Unterricht der analytischen Geometrie wird in Abschnitt 1.4.6 der vorliegenden Arbeit eingegangen.

### 1.4.2 Kalkülorientierung und Dominanz von Routineaufgaben

Nachdem die im Zuge der „Neuen Mathematik“ eingeführte formale Behandlung abstrakter algebraischer Strukturen im Mathematikunterricht weitgehend an Bedeutung verloren hat, ist m. E. die Fokussierung des Unterrichts auf „Aufgabeninseln“ und das (allzu oft) mechanische Einüben entsprechender Routineaufgaben das gravierendste Problem des heutigen Mathematikunterrichts,<sup>66</sup> das im Unterricht der analytischen Geometrie besonders auffällig zutage tritt.

Der auf S. 25 skizzierte Aufbau einer vektoriellen Raumgeometrie, der die Rahmenpläne der meisten Bundesländer im Stoffgebiet Analytische Geometrie bestimmt, ermöglicht eine Einteilung des Unterrichts in (hauptsächlich) die folgenden „Aufgabeninseln“:

- Vektorielle Beweise einfacher geometrischer Sätze;
- Umwandlung von Parameterdarstellungen von Geraden und Ebenen in Koordinatengleichungen bzw. Normalengleichungen und umgekehrt;
- Lageuntersuchungen und Schnittmengenberechnungen (Gerade–Gerade, Gerade–Ebene, Ebene–Ebene) mithilfe des GAUSS-Algorithmus (Spezialfälle: Bestimmung von Spurpunkten bzw. -geraden);
- Berechnung von Abständen (Gerade–Gerade, Punkt–Ebene) unter Anwendung des Skalar- und z. T. des Vektorproduktes;
- Berechnung von Schnittwinkeln (Gerade–Gerade, Gerade–Ebene, Ebene–Ebene).<sup>67</sup>

Eine der Ursachen für die starke Orientierung des Unterrichts auf das Einüben von Kalkülen und das Lösen von Routineaufgaben besteht sicherlich darin, dass sich das Ausführen von Kalkülen (auch in Ländern ohne Zentralabitur) gut für die Gestaltung von Abituraufgaben eignet, die für die Schüler in zweifacher Hinsicht „berechenbar“

---

<sup>66</sup>Es bedarf nicht einmal der zum Beleg der Ausrichtung des deutschen Mathematikunterrichts an Routineaufgaben oft herangezogenen PISA-Ergebnisse, um auf diesen Missstand aufmerksam zu werden. Erschreckend sind Berichte von Studierenden über ihren Mathematikunterricht in der gymnasialen Oberstufe, der sich vielfach hauptsächlich mit dem Einüben von Aufgabentypen befasste, die im schriftlichen Abitur zu erwarten waren. Noch erschreckender ist das Ergebnis. Häufig haben Studierende bereits nach wenigen Jahren fast alles vergessen, was sie in der analytischen Geometrie gelernt hatten. Ohne quantitative Belege anführen zu können, ist m. E. zu vermuten, dass die Mehrzahl der Studierenden in ihrer Schulzeit z. B. keine inhaltliche Vorstellung von der Beschreibung einer Geraden oder einer Ebene durch eine Parameterdarstellung erworben, die Idee der Parametrisierung geometrischer Objekte also nicht verstanden hat. Eine große Zahl von Aufgaben, in denen Parameterdarstellungen von Geraden und Ebenen auftreten, kann an dieser Situation nichts ändern, wenn bei der Behandlung der Aufgaben das Einüben von Lösungsschemata im Vordergrund steht.

<sup>67</sup>In Abhängigkeit vom Rahmenplan kommen noch Lageuntersuchungen zwischen Geraden bzw. Ebenen und Kugeln hinzu. G. SCHMIDT kritisierte bereits 1993 die Einheitlichen Prüfungsanforderungen für das Abitur (EPA) von 1989 dafür, dass sie Aufgaben der aufgeführten Kategorien (vor allem für das Abitur in Grundkursen) nach sich ziehen, deren Lösungswege rein algebraischer Natur sind und ohne geometrische Überlegungen bzw. Veranschaulichungen auskommen (vgl. [218], S. 16f. und [276], S. 65). Obwohl 2002 neue EPA in Kraft traten, hat sich an den Abituraufgaben im Bereich der analytischen Geometrie bislang wenig geändert. Dies ist nicht verwunderlich, da die vektorielle analytische Geometrie des Raumes mit den auf S. 25 genannten Schwerpunkten nach wie vor die Rahmenpläne dominiert.



sind. In Ländern mit Zentralabitur haben die zu erwartenden Aufgabentypen (die sich zumindest nach dem heutigen Stand innerhalb einer gewissen Schwankungsbreite gut vorhersehen lassen) ohnehin vielfach die Stellung eines „inoffiziellen Lehrplans“.<sup>68</sup>

In der Expertise [29] wird die Auffassung vertreten, dass „*die Konzentration auf Kalküle und Routinen wohl vor allem als Strategie des Lehrers anzusehen (ist), um trotz Schwierigkeiten und teils geringer Anstrengungsbereitschaft bei Lernenden Sicherheit in der Beherrschung der mathematischen Methoden zu erreichen*“. Besorgnis erregend ist vor allem ein damit einhergehender „Erziehungseffekt“, der viele Schüler (auch solche, die über eine hohe Anstrengungsbereitschaft verfügen) zu der Überzeugung führt, Mathematik ließe sich durch das intensive „Training“ von Aufgabentypen erlernen.<sup>69</sup>

Die starke Kalkülorientierung des Unterrichts führt dazu, dass anschauliche Vorstellungen und inhaltliches Verständnis bei den Schülern unzureichend ausgeprägt werden.<sup>70</sup> Die von TIETZE konstatierte geringe „Transferbreite formaler Sachverhalte, die in der Schule gelernt wurden“ (vgl. S. 20) trifft m. E. auf eingeübte Standardaufgaben in ähnlichem Maße zu und die Vergessensrate ist hoch (vgl. Fußnote 66 auf S. 34).

Die Überwindung der einseitigen Fixierung des Unterrichts auf das Lösen von Routineaufgaben erfordert Ansätze auf mehreren Gebieten. SCHUPP begründete 2000, dass die curriculare Situation, also die Behandlung einer vektoriellen analytischen Geometrie mit den auf S. 25 genannten Schwerpunkten und die dadurch bedingte Formenarmut des Unterrichts mit einer Kalkülorientierung einhergeht und forderte die Exploration interessanter geometrischer Objekte. Dabei verbessert die Auswahl der Objekte allein die Situation noch nicht, wie er am Beispiel des „Durchexerzierens der Kegelschnitte“ deutlich machte ([234], S. 62). Die Herausarbeitung fundamentaler Ideen im Unterricht der analytischen Geometrie ist ebenfalls eine notwendige Bedingung dafür, die Erstarrung in Routineaufgaben aufzulösen. Notwendig erscheint dafür aber auch eine Unterrichtsgestaltung, die selbstständigem Lernen der Schüler mit offeneren Zielsetzungen mehr Raum zuzusst. Schließlich kann ein experimentelles, modellierendes und in stärkerem Maße die Anschauung einbeziehendes Arbeiten die einseitige Fixierung auf das Abarbeiten formaler Kalküle aufbrechen helfen. Die Einbeziehung der 3D-Computergrafik schafft hierfür, wie in dieser Arbeit aufgezeigt werden soll, in Bezug auf die räumliche analytische Geometrie reichhaltige Ansatzpunkte und Möglichkeiten.

---

<sup>68</sup>Sehr populär sind Sammlungen von Abituraufgaben früherer Jahre (wie z. B. [5]) sowie „Trainingsbücher“ wie [104], die das Einüben der Aufgaben erleichtern sollen.

<sup>69</sup>Die Vorstellung, Mathematik durch das Lernen von Fakten und das Lösen hinreichend vieler Aufgaben nach vorgegebenen Schemata zu „trainieren“, ist erfahrungsgemäß auch unter Lehramtsstudierenden weit verbreitet und birgt natürlich die Gefahr der „Vererbung“ auf künftige Schülergenerationen in sich. Der Abbau derartiger Vorstellungen gehört deshalb zu den wichtigsten Aufgaben des Lehramtsstudiums und kann nur durch das aktive Erleben mathematischer Ideen, das Lösen von Problemen, logisches Schlussfolgern und eigenständiges mathematisches Modellieren gelingen.

<sup>70</sup>Bereits H. LENNÉ bemängelte, dass Mathematik den Lernenden hauptsächlich, teilweise sogar fast ausschließlich, als Sammlung von Aufgabentypen entgegentritt (vgl. [154], S. 34–37 und 50–54); diese Situation besteht vielfach immer noch (siehe z. B. [234], S. 58).

### 1.4.3 Grundvorstellungen und fundamentale Ideen

Die Diskussion um „fundamentale Ideen“, „Leitideen“ bzw. „universelle Ideen“, die eine Orientierungsgrundlage für den Mathematikunterricht bilden, hat durch die Entwicklung von Bildungsstandards an Beachtung gewonnen.<sup>71</sup> Die in den Bildungsstandards und den EPA festgehaltenen Leitideen können jedoch nur eine grobe Orientierung für die Behandlung einzelner Stoffgebiete geben;<sup>72</sup> es ist daher eine Präzisierung anhand der spezifischen Inhalte und Methoden der jeweiligen mathematischen Teilgebiete erforderlich. Um eine derartige Präzisierung vorzunehmen, analysierte TIETZE in [244] und [247] fundamentale Ideen der analytischen Geometrie / linearen Algebra und gliederte sie in die drei Kategorien *Leitideen*,<sup>73</sup> *zentrale Mathematisierungsmuster* und *bereichsspezifische Strategien*.<sup>74</sup> Für den Mathematikunterricht besonders bedeutsame Elemente aller drei Kategorien werden in der didaktischen Diskussion mitunter als „zentrale Ideen“ bezeichnet, wobei nach KLIKA den Mathematisierungsmustern eine besondere Bedeutung zukommt, da sie „gewissermaßen die Katalysatoren beim Mathematisieren“ sind ([134], S. 6). Dass eine „scharfe“ Trennung von Inhalten, Strategien und Mathematisierungsmustern nicht in jedem Falle sinnvoll ist, wird z. B. anhand der Idee des funktionalen Zusammenhangs deutlich, die sich allen drei Kategorien zuordnen lässt.

TIETZE arbeitete auf der Grundlage einer fachgebietsimmanenten sowie anwendungsorientierten Analyse Leitideen, zentrale Mathematisierungsmuster und bereichsspezifische Strategien der linearen Algebra und analytischen Geometrie heraus (vgl. [247], S. 2–68).

<sup>71</sup>Entsprechende Konzepte wurden bereits 1929 von WHITEHEAD entwickelt und sind in der Mathematikdidaktik seit langem von Bedeutung. So wurde z. B. in der DDR der Mathematikunterricht durch „Lehrplanleitlinien“ längsschnitthaft gegliedert.

<sup>72</sup>Die Einheitlichen Prüfungsanforderungen für das Abitur von 2002 (EPA, siehe [136]) gliedern die fachlichen Inhalte nach den Leitideen *Funktionaler Zusammenhang*, *Grenzprozesse/Approximation*, *Modellieren*, *Messen*, *Algorithmus*, *Räumliches Strukturieren/Koordinatisieren* sowie *Zufall*. Demgegenüber lehnen sich die ebenfalls nach Leitideen gegliederten neuen Bildungspläne für das Gymnasium in Baden-Württemberg (vgl. S. 30) an die Bildungsstandards der KMK für den mittleren Bildungsabschluss an, erweitern diese jedoch; es finden sich hier die Leitideen *Zahl*, *Variable*, *Algorithmus*, *Messen*, *Raum und Form*, *Funktionaler Zusammenhang*, *Daten und Zufall*, *Vernetzung* sowie *Modellieren*. In beiden Varianten werden inhalts- und prozessbezogene Ideen nicht voneinander getrennt, was eine Abweichung vom Aufbau der amerikanischen NCTM-Standards bedeutet, die ansonsten einen erkennbaren Einfluss auf die KMK-Standards hatten.

<sup>73</sup>Der Begriff „Leitideen“ wurde von TIETZE somit in einem anderen Sinne gebraucht als in den Bildungsstandards und den EPA. Zu den dort aufgeführten allgemeinen Leitideen analoge schulstufen- und stoffgebietsübergreifende Konzepte bezeichnete er als „universelle Ideen“ (siehe auch [246]).

<sup>74</sup>TIETZE charakterisierte diese Kategorien folgendermaßen ([247], S. 2):

- „*Leitideen* sind Begriffe und Sätze, die innerhalb des Implikationsgefüges einer mathematischen Theorie eine zentrale Bedeutung haben, indem sie gemeinsame Grundlage zahlreicher Aussagen dieser Theorie sind und/oder einem hierarchischen Aufbau dienen.
- *Zentrale Mathematisierungsmuster* repräsentieren den außermathematischen Verwendungs- und Modellierungsaspekt eines mathematischen Gebietes und stehen in enger Beziehung zu dem curricularen Gesichtspunkt Erfordernisse des tertiären Bereichs.
- *Bereichsspezifische Strategien* spielen im Rahmen des Problemlösens und Begriffsbildens eine wichtige prozessbezogene Rolle.“

Basierend auf der dadurch erfolgten Charakterisierung des Fachgebietes und unter Heranziehung universeller Ideen des Mathematikunterrichts formulierte er „fundamentale Ideen zur Analytischen Geometrie und Linearen Algebra im Umfeld der Schulmathematik“ ([247], S. 71). Diese umfassen verschiedene (in 1.2.4 und 1.3 skizzierte) Ausrichtungen des Stoffgebietes. Da die vorliegende Arbeit auf eine Behandlung der analytischen Geometrie des Raumes unter Betonung geometrischer Inhalte und Anwendungen zielt, erfolgt hier eine Beschränkung (bei gleichzeitiger Präzisierung) auf in diese Richtung weisende Aspekte. Grundvorstellungen<sup>75</sup> und Strategien hinsichtlich der folgenden fundamentalen Ideen der analytischen Geometrie des Raumes sind m. E. unverzichtbar:<sup>76</sup>

1. Beschreibung von Punkten (im Sinne räumlicher Positionen) durch Koordinaten verbunden mit „koordinatengebundener Anschauung“ (d. h. sinnlicher Vorstellungen des Einflusses von Koordinatenwerten auf die Lage im Raum, siehe 1.4.4).

Eine die Anschauung unterstützende Ergänzung hierzu kann die Beschreibung der Lage einfacher Körper durch Koordinaten charakteristischer Punkte sein.

2. Beschreibung geometrischer Objekte (Kurven und Flächen mit den Spezialfällen Geraden und Ebenen) durch Gleichungen; Verständnis von Gleichungen als Bedingungen an Punkte bzw. Koordinatentripel, denen jeweils Teilmengen aller Punkte des Raumes genügen (welche die geometrischen Objekte darstellen); Transfer zwischen Gleichungen als algebraisch formulierten Bedingungen und der geometrischen Interpretation ihrer Lösungsmengen als Punktmengen.

Anstrebenswert ist in diesem Zusammenhang die Herausbildung eines intuitiven „Dimensionsverständnisses“ in dem Sinne, dass Bedingungen in Form von Gleichungen die Zahl der „Ausdehnungsrichtungen“ der durch sie beschriebenen Objekte einschränken.

3. Geometrisches und elementares algebraisches Verständnis für Schnittmengen von Objekten als Grundlage geometrischer Anwendungen der linearen Algebra.

Ein solches Grundverständnis beinhaltet, dass die Schüler mit der Verwendung linearer Gleichungssysteme für Schnittberechnungen inhaltliche Vorstellungen verbinden. Es ist auch eine Voraussetzung dafür, die Beschreibung von Geraden im Raum durch 2 lineare Gleichungen zu

---

<sup>75</sup>Der Begriff „Grundvorstellungen“ wird hier im Sinne der von VOM HOFÉ in [118], S. 97f. gegebenen Charakterisierung verwendet: „Die Grundvorstellungs-idee beschreibt Beziehungen zwischen mathematischen Inhalten und dem Phänomen der individuellen Begriffsbildung. In ihren unterschiedlichen Ausprägungen charakterisiert sie ... insbesondere drei Aspekte dieses Phänomens:

- Sinnkonstituierung eines Begriffs durch Anknüpfung an bekannte Sach- oder Handlungszusammenhänge bzw. Handlungsvorstellungen,
- Aufbau entsprechender (visueller) Repräsentationen bzw. ‚Verinnerlichungen‘, die operatives Handeln auf der Vorstellungsebene ermöglichen,
- Fähigkeit zur Anwendung eines Begriffs auf die Wirklichkeit durch Erkennen der entsprechenden Struktur in Sachzusammenhängen oder durch Modellieren des Sachproblems mit Hilfe der mathematischen Struktur.“

Zum Verhältnis von fundamentalen Ideen und Grundvorstellungen führte VOM HOFÉ u. a. aus: „Eine ‚fundamentale Idee‘ läßt sich in zahlreichen als normative Kategorien verstandenen Grundvorstellungen konkretisieren, jeder solchen normativen Kategorie entspricht schließlich eine Fülle individueller Erklärungsmodelle“ ([118], S. 128f.).

<sup>76</sup>Recht ähnliche Aspekte arbeitete WITTMANN in [277] als zentrale Ideen der analytischen Geometrie heraus.

verstehen. In diesem Zusammenhang kann das bereits erwähnte intuitive Dimensionsverständnis dadurch vertieft werden, dass mehrere Bedingungen die Dimension der Punktmengen, die ihnen genügen, reduzieren. Wird der GAUSS-Algorithmus behandelt, so ist dessen geometrische Interpretation sinnvoll.

4. Analogiebetrachtungen zwischen ebenen und räumlichen geometrischen Objekten und Sachverhalten; Rückführung räumlicher auf ebene Probleme als fundamentale Vereinfachungsstrategie; Betrachtung von Schnittfiguren mit Koordinatenebenen.

Die mit diesen vier Aspekten verbundenen mathematischen Inhalte sind Bestandteile der meisten gegenwärtigen Kurse zur analytischen Geometrie und linearen Algebra, werden aber, wie bereits ausgeführt wurde, oft formal abgehandelt. Daher ist die Betonung auf die Herausbildung entsprechender Grundvorstellungen durch die Schüler zu legen. Dies lässt sich nicht durch das alleinige Lösen einer hinreichend großen Anzahl von Rechenaufgaben erreichen, sondern bedarf sorgfältiger Überlegungen, Veranschaulichungen sowie heuristischer Herangehensweisen. Insbesondere hinsichtlich des Punktes 2. sei hinzugefügt, dass im Sinne des Betreibens wirklicher „Geometrie“ keine Beschränkung auf lineare Gleichungen und somit Geraden und Ebenen erfolgen sollte. Auch Schnittbetrachtungen nichtlinearer Objekte sind anstrebenswert.

Wird eine vektorielle analytische Geometrie betrieben, so ist es über die Aspekte 1.–4. hinaus erforderlich, Grundvorstellungen hinsichtlich mit dem Vektorbegriff verbundener Ideen herauszubilden:

5. Vektoren als abstrahierende Objekte, die verschiedene Sachverhalte beschreiben und miteinander in Verbindung bringen, wie Positionen im Raum, Verschiebungen und Kräfte;<sup>77</sup> Pfeile als Repräsentanten von Vektoren; Vektoren als  $n$ -Tupel.
6. Vereinfachung einiger Darstellungen und Berechnungen in der analytischen Geometrie durch die Verwendung von Vektoren; Vektorrechnung als geometrische Operationen beschreibender Kalkül.
7. Vektorielle Parameterdarstellungen von Geraden und Ebenen als Beschreibungen dieser Gebilde durch Punkte und Ausdehnungsrichtungen.

Im Zusammenhang mit Parameterdarstellungen ist eine Ergänzung durch den folgenden Aspekt unbedingt wünschenswert:

8. Geraden und Ebenen (sowie Kurven und Flächen) als Punktmengen; Parameterdarstellungen als Funktionen der Koordinaten von Punkten des Raumes in Abhängigkeit von reellen Zahlen.

Diese Auffassung von Parameterdarstellungen, auf die in der vorliegenden Arbeit noch näher eingegangen wird, ermöglicht sowohl die parameterabhängige Beschreibung interessanter Kurven und Flächen als auch „kinematisch“ geprägte Vorstellungen von Kurven als Bewegungsbahnen. Sie ist nicht an die Verwendung von Vektoren gebunden und kann auch vektorfrei entwickelt werden. Bei Parameterbeschreibungen von Geraden und Ebenen stellt sich der Nutzen der Verwendung von Vektoren durch die Vereinfachung von Schreibweisen und Rechnungen heraus.

---

<sup>77</sup>Ein interessantes Beispiel aus der Computergrafik ist die Beschreibung von Farben durch Vektoren. Auf die Behandlung von Vektoren, das Verhältnis arithmetischer und geometrischer Aspekte sowie die Verwendung des Vektorbegriffs in der Informatik wird in Abschnitt 3.3.1 näher eingegangen.

Schließlich seien noch fundamentale Ideen im Zusammenhang mit der metrischen Geometrie des Raumes genannt:

9. Skalarprodukt als Grundlage einer Metrik des Anschauungsraumes im Sinne einer analytischen Beschreibung von Längen- und Winkelmessungen.
10. Metrische Beziehungen als Grundlage der Beschreibung geometrischer Objekte.  
Beispiele hierfür sind die Beschreibungen von Kreisen und Kugeln durch Abstände sowie Normalendarstellungen von Geraden und Ebenen.

Dass die metrischen Aspekte in dieser Aufzählung am Ende stehen, soll nicht implizieren, dass sie erst am Ende des Stoffgebietes behandelt werden sollten. Eine Trennung zwischen affiner und metrischer Geometrie würde von der Mehrzahl der Schüler als künstlich empfunden werden, da sie hierfür über keine ausreichende Abstraktionsbasis verfügen. Vielmehr können Abstände von Punkten bereits vor der Einführung von Vektoren und auch bei einer ausschließlich koordinatengeometrischen Behandlung des Stoffgebietes betrachtet werden, wobei es nahe liegt, an den Satz des Pythagoras als zentrale metrische Beziehung des Geometrieunterrichts der Sekundarstufe I anzuknüpfen.<sup>78</sup>

### Zusammenfassung

Es wurden in diesem Abschnitt einige grundlegende Ideen und damit im Zusammenhang bei den Schülern zu entwickelnde Vorstellungen genannt, die m. E. bei der Behandlung einer vektoriellen analytischen Geometrie unverzichtbar sind. Abstriche an einem tiefgehenden Verständnis dieser Ideen zugunsten der Behandlung zusätzlicher mathematischer Inhalte erscheinen nicht akzeptabel, denn dies hieße, auf nicht vorhandene Fundamente zu bauen. Umgekehrt sind eher Streichungen an den mit den aufgeführten Stichpunkten verbundenen Inhalten hinzunehmen (evtl. sogar hinsichtlich der Behandlung von Vektoren), wenn die dadurch zu gewinnende Zeit für die Herausbildung eines fundierten Verständnisses der grundlegenden (koordinatenbezogenen) Arbeitsweisen der analytischen Geometrie benötigt wird.

Auf der Grundlage gefestigter Vorstellungen der Schüler hinsichtlich der aufgeführten Aspekte sind vor allem in Leistungskursen Erweiterungen sowohl hinsichtlich struktureller algebraischer Überlegungen<sup>79</sup> als auch der behandelten geometrischen Objekte und der dabei angewendeten Betrachtungsweisen möglich. Verwiesen sei dazu auf die folgenden Abschnitte und auf von TIETZE herausgestellte fundamentale Ideen in Bezug auf die analytische Beschreibung geometrischer Abbildungen und die Modellierung nichtgeometrischer Sachverhalte ([247], S. 71) sowie auf seine Ausführungen hinsichtlich in Grund- und Leistungskursen sinnvoll zu behandelnder Inhalte (ebd., S. 154–158).

---

<sup>78</sup>Auch bei der Behandlung des Skalarproduktes erscheint eine Bezugnahme auf Vorkenntnisse aus der SI (Satz des Pythagoras, Kosinussatz) sinnvoll, näher wird darauf in Abschnitt 4.6.2 eingegangen.

<sup>79</sup>So bietet z. B. der Dimensionsbegriff, nachdem die Schüler dafür ein intuitives Verständnis erworben haben, Möglichkeiten für theoretische Vertiefungen und Exaktifizierungen.

#### 1.4.4 Anschauung und Raumvorstellung im Unterricht der analytischen Geometrie

Raumanschauung, Raumvorstellung und Veranschaulichung werden in der Mathematikdidaktik oft – sowohl als Ziel als auch als Mittel der Erkenntnisgewinnung – genannt.<sup>80</sup> Häufig erfolgt dabei keine präzise Beschreibung und Abgrenzung dieser Begriffe gegeneinander, was sicherlich damit zusammenhängt, dass gemeinsame Gesichtspunkte und Überlappungen auftreten. Um zur Bedeutung anschaulicher Vorstellungen für den Unterricht in analytischer Geometrie und den diesbezüglichen Potenzialen dieses Stoffgebietes vorzudringen, erscheint zunächst eine Unterscheidung zweier Aspekte angebracht:

- (1) Veranschaulichung ursprünglich nicht geometrischer Objekte und Zusammenhänge, die in einer bildlichen bzw. geometrischen Form interpretiert werden;<sup>81</sup>
- (2) geometrische Anschauung im Sinne von Raumvorstellung als geistige Repräsentation bereits ihrem Ursprung nach geometrischer Objekte und Beziehungen.<sup>82</sup>

Beide Bereiche können natürlich miteinander kombiniert auftreten. Auch die geometrische Interpretation komplexer mathematischer Zusammenhänge bedarf der individuellen Verinnerlichung durch den Aufbau von Vorstellungen, um eine veranschaulichende Wirkung zu erzielen. (1) und (2) stellen somit zwei Stufen der Veranschaulichung mathematischer Objekte und Zusammenhänge dar, von denen die erste entfällt, wenn es sich bereits um unmittelbar geometrische Sachverhalte handelt.

Die analytische Geometrie ist ihrem Wesen nach eine veranschaulichende Disziplin in dem Sinne, dass sie die Möglichkeit der geometrischen Interpretation arithmetischer und algebraischer Sachverhalte schafft. VOLKERT arbeitete zwar heraus, dass ein Ziel des auf der Koordinatenmethode beruhenden Arithmetisierungsprogramms darin bestand, die Anschauung aus der Mathematik herauszudrängen ([253], S. XX):

*„Das Arithmetisierungsprogramm beruht auf der Koordinatenmethode DESCARTES', die den erforderlichen Zusammenhang zwischen Geometrie und Analysis herstellt. Damit schien es*

<sup>80</sup>Bereits im Zuge der Meraner Reformen wurde der Entwicklung des räumlichen Anschauungsvermögens eine hohe Bedeutung zugemessen, siehe 1.2.1 und [121], S. 159–169. Einen Überblick über die Anschauung als „Quelle neuen Wissens“ gab WINTER in [273], Kap. 8; er bezog sich u. a. auf die Arbeit [253] von VOLKERT, die eine umfassende Diskussion des Begriffs der Anschauung und der Bedeutung anschaulicher Vorgehensweisen und Begründungen in der Geschichte der Mathematik enthält.

<sup>81</sup>Als Beispiele hierfür seien Veranschaulichungen von Brüchen sowie die geometrische Interpretation der komplexen Zahlen in der GAUSSschen Zahlenebene genannt. Wie u. a. VOLKERT und WINTER ausführten, gab die Veranschaulichung mithilfe der GAUSSschen Zahlenebene sogar für die „Anerkennung“ der komplexen Zahlen den Ausschlag, siehe [253], S. 33–46 und [273], S. 148–152.

<sup>82</sup>Es wird hier nicht auf die philosophische Frage nach der Realität des Raumes eingegangen, der als transzendentaler Natur im Sinne KANTS (als allen Menschen übereinstimmend gegebener Komplex von Empfindungen) oder als reale, auch außerhalb menschlicher Vorstellungen existierende Kategorie aufgefasst werden kann (zu dieser Auffassung gelangte u. a. GAUSS, siehe z. B. [65], S. 163f.). Für die in dieser Arbeit anzustellenden Überlegungen kommt es lediglich darauf an, dass der euklidische „Anschauungsraum“ objektiv in dem Sinne existiert, dass Kommunikation darüber möglich ist, verschiedene Menschen also gleichartige Vorstellungen herausbilden. Diese Voraussetzung ist auf der Grundlage beider Auffassungen des Begriffs „Raum“ gegeben.

*möglich, den ureigensten Bereich der Anschauung – die Geometrie nämlich – auf den als unproblematischer angesehen Bereich der reellen Zahlen und ihrer Funktionen zurückzuführen.“*

Jedoch besteht hierin nur scheinbar ein Widerspruch zu der These, dass die analytische Geometrie eine veranschaulichende Disziplin ist; die Verbindung, welche sie zwischen Geometrie sowie Arithmetik, Algebra und Analysis herstellt, ist bidirektional<sup>83</sup> und kann deshalb sowohl zur Algebraisierung (und somit Formalisierung) geometrischer Objekte und Beziehungen als auch zur Geometrisierung algebraischer Sachverhalte genutzt werden (siehe auch [253], S. 22ff.) und damit ihre Veranschaulichung erschließen.<sup>84</sup>

Veranschaulichungsvoraussetzungen im Sinne des auf S. 40 genannten Aspekts (1) sind bei der Behandlung der analytischen Geometrie also a priori gegeben, jedoch wird dadurch noch lange nicht „automatisch“ das Anschauungsvermögen der Schüler (weder im engeren Sinne der räumlichen Anschauung noch im weiteren Sinne der Nutzung anschaulicher Vorstellungen als heuristisches Mittel der Erkenntnisgewinnung) entwickelt. Wie TIETZE in [247], S. 116 begründete, besteht ein derartiger Automatismus generell, auch bei der Beschäftigung mit Elementargeometrie, nicht:

*„Die Raumanschauung wird nur dann gefördert, wenn sie gezielt, insbesondere durch Kopfgeometrie und unterstützendes Arbeiten mit Modellen und Materialien, angestrebt wird.“*

Aktives, geistiges und konkret-handelndes Operieren ist eine Grundvoraussetzung für die Entwicklung des Anschauungs- bzw. Vorstellungsvermögens, vor allem im Sinne des auf S. 40 aufgeführten Aspekts (2). Für die Einbeziehung von Elementen der Computergrafik in den Unterricht ergibt sich daraus die (ohnein nahe liegende) Schlussfolgerung, dass die bloße Nutzung von Produkten wie Bildern und Videos kaum zur Entwicklung der Raumanschauung beitragen wird und Visualisierungen hierzu nur dann einen nennenswerten Beitrag liefern können, wenn eine aktive Auseinandersetzung damit erfolgt bzw. die Schüler diese selbst erstellen. CUNNINGHAM legte auf diesen Aspekt bei der Nutzung von Computervisualisierungen im Mathematikunterricht großen Wert:

*„However, to make the best use of visualization in mathematics instruction, it is necessary to make it an active part of the instructional process: students must actively do visual mathematical activities, not just observe mathematical visuals.“* ([42], S. 85)

Die Entwicklung des *räumlichen Vorstellungsvermögens* (im engeren Sinne der dreidimensionalen Raumvorstellung) wird häufig als Ziel des Unterrichts in analytischer Geometrie genannt. Nach THURSTONE (1938) ist die Raumvorstellung (neben dem Sprach-

---

<sup>83</sup>Diese Bidirektionalität wird bereits anhand der auf S. 10 aufgeführten Hauptaspekte der frühen Koordinatengeometrie deutlich.

<sup>84</sup>Durch die Interpretation algebraischer Objekte schafft die analytische Geometrie auch verallgemeinerte „Geometrien“ wie  $n$ -dimensionale Räume, die zwar der unmittelbaren sinnlichen Anschauung nicht zugänglich sind, durch Analogiebetrachtungen aber mit anschaulichen Vorstellungen verknüpft werden können. Sicherlich ist eine unmittelbare sinnliche Vorstellung von einem  $m$ -dimensionalen Teilraum eines  $n$ -dimensionalen affinen Raumes schwerlich zu erlangen; die Analogie zu ein- und zweidimensionalen Teilräumen dreidimensionaler Räume ermöglicht aber die Verknüpfung mit anschaulichen Vorstellungen, was neben veranschaulichenden auch abstrahierende Überlegungen erfordert.

verständnis, der Wortflüssigkeit, dem Umgang mit Zahlen, der assoziativen Leistung des Gedächtnisses, dem schlussfolgernden Denken und der Auffassungsgeschwindigkeit) eine der sieben primären intellektuellen Fähigkeiten und umfasst 3 Teilfähigkeiten:

- *Räumliches Orientieren* als Fähigkeit, sich wirklich oder gedanklich im Raum orientieren zu können;
- *Räumliches Vorstellen* als Fähigkeit, Objekte oder Beziehungen in der Vorstellung reproduzieren zu können;
- *Räumliches Denken* als Fähigkeit, mit räumlichen Vorstellungsinhalten gedanklich zu operieren.

Mehrfach wurde festgestellt, dass die Entwicklungsfähigkeit der Raumvorstellung in diesem unmittelbaren Sinne im Grundschulalter (7.–11. Lebensjahr) sehr hoch ist, danach abnimmt und bereits mit 17 Jahren nur noch recht geringe Zuwächse möglich sind (vgl. u. a. [20], S. 2f.). Hinsichtlich der Entwicklung der Raumvorstellung als primärer Fähigkeit bestehen demnach in der Sekundarstufe II nur sehr begrenzte Möglichkeiten.

Bei der Behandlung der analytischen Geometrie kann es m. E. jedoch auch nicht vorrangiges Ziel sein, die unmittelbare Raumvorstellung zu entwickeln. Vielmehr sollte es aufgrund der Inhalte dieses Sachgebietes um die Herausbildung einer „vermittelten“ Raumvorstellung gehen, d. h. der Fähigkeit, Positionen und Objekten im Raum Zahlen (Koordinaten) bzw. sogar Gleichungen zuzuordnen und umgekehrt mit Koordinaten und Gleichungen anschauliche (räumliche) Vorstellungen zu verbinden.<sup>85</sup> Insofern sind die im Unterricht der analytischen Geometrie entwickelbaren Komponenten der Raumanschauung bzw. -vorstellung durch die in den EPA 2002 verwendeten Begriffe „Koordinatisieren / räumliches Strukturieren“ (siehe S. 36)<sup>86</sup> besser erfasst als durch den Begriff „Raumvorstellung“, denn diese Begriffe verbinden räumliche Vorstellungen mit arithmetischen und algebraischen Lage- bzw. Objektbeschreibungen.

### **Koordinatenbezogene Raumvorstellung**

Es sind mir keine Untersuchungen bekannt, die sich auf die Herausbildung einer „koordinatenvermittelten Raumvorstellung“ beziehen, weshalb die folgenden Überlegungen etwas spekulativ bleiben müssen und lediglich auf Plausibilitätsbetrachtungen, Eigenbeobachtungen und Gesprächen mit einer recht kleinen Zahl von Schülern und Studierenden basieren. Lernpsychologisch fundierte Untersuchungen zu dieser Thematik wären

---

<sup>85</sup>Natürlich ist dafür ein Mindestmaß an bereits vorhandener primärer Raumvorstellung unabdingbar und eine Entwicklung derselben wird bei der aktiven Veranschaulichung von durch Koordinaten bzw. Gleichungen gegebenen geometrischen Objekten sicherlich in einem gewissen Maße eintreten. Die o. g. drei Teilfähigkeiten der Raumvorstellung bieten auch in einem vermittelten (wenngleich von THURSTONE sicherlich nicht gemeinten) Sinne Anhaltspunkte für Komponenten einer „koordinatenvermittelten Raumvorstellung“ – insbesondere dann, wenn auch die „eigene“ (betrachtende) Position durch Koordinaten beschrieben werden muss. Dies ist bei der Arbeit mit einigen Grafik-Softwarepaketen, auf die in dieser Arbeit noch ausführlicher eingegangen wird, der Fall.

<sup>86</sup>Allerdings werden diese Leitideen in den EPA nicht anhand von Vorstellungen beschrieben, sondern durch einige zentrale Inhalte des Stoffgebietes Lineare Algebra / Analytische Geometrie und darauf bezogene Fähigkeiten bestimmt (vgl. [136], S. 8).



m. E. ausgesprochen wünschenswert, sie würden jedoch erheblich über den Rahmen der vorliegenden Arbeit hinausgehen.

Eine Möglichkeit des handlungsorientierten und visuell unterstützten Umgangs mit Koordinatentripeln und den zugehörigen Positionen im Raum besteht in der Konstruktion von zusammengesetzten Objekten durch Koordinatenbeschreibungen der einzelnen Bestandteile mithilfe einer geeigneten Software.<sup>87</sup> Dies erfordert das gedankliche Operieren mit räumlichen Objekten, die Zuordnung von Koordinaten und die Überprüfung der Ergebnisse durch Veranschaulichung in der verwendeten Software. Danach können durch Koordinatenänderungen Unstimmigkeiten beseitigt werden, so dass eine „passgerechte“ Konstruktion entsteht. Die Visualisierung mithilfe der Software hat in diesem Prozess zwar eine wichtige Bedeutung; unabdingbar ist jedoch der gedankliche Transfer zwischen der gewünschten Lage im Raum und den dazu notwendigen Koordinatenwerten. Dies erfordert die gedankliche Einteilung des Raumes in grobe, durch Koordinatenintervalle beschriebene Bereiche sowie die Zuordnung von Koordinatenänderungen zu unten-oben-, links-rechts- und vorn-hinten-Verschiebungen.

Entscheidend für die Erlangung einer „koordinatenbezogenen Raumvorstellung“ sind die notwendigen Vorüberlegungen hinsichtlich der Zuordnung von Koordinaten zu Positionen und nur in zweiter Linie die Betrachtung der berechneten Abbildungen. Allerdings sind beide Aspekte nicht voneinander zu trennen, da die Visualisierung eine Rückmeldung hinsichtlich der Koordinatisierungsüberlegungen gibt und zugleich den Anlass für Änderungen der Koordinaten liefert. Häufig stellten Schüler und Studierende fest, dass die anfänglichen Überlegungen zur Positionierung von Objekten den schwierigsten Teil der Arbeit darstellten, sie dann aber – auch durch Versuch-und-Irrtum-Vorgehensweisen – ein „Gefühl“ für die Beziehung zwischen Koordinaten- und Positionsänderungen erlangten und begannen, koordinatenbezogen zu denken.

Es liegt nahe, dass die Fähigkeit des Koordinatisierens Transfers ermöglicht, mittels derer Stärken in bestimmten Bereichen geistigen Operierens zur Kompensation von Schwächen in anderen Bereichen genutzt werden können. Für Personen mit einer gut entwickelten Fähigkeit der gedanklichen Veranschaulichung mathematischer Objekte und Relationen, die nicht geometrischen Ursprungs sind, kann die Koordinatisierung als Kompensationsstrategie für eine geringer ausgebildete unmittelbare Raumvorstellung nützlich sein – mitunter lassen sich Koordinatensysteme in räumliche Situationen „hinein denken“, um die Orientierung zu erleichtern. Für Personen mit einer gut ausgeprägten Raumvorstellung kann die durch das Koordinatisieren erfolgende Abbildung zwischen Zahlen und räumlichen Positionen dabei helfen, die primäre Raumvorstellung zu nutzen, um Vorstellungen mathematischer Sachverhalte zu entwickeln.

---

<sup>87</sup>Auf entsprechende Beispiele sowie Ergebnisse von Schülern und Studierenden unter Verwendung der Software POV-Ray wird in Abschnitt 4.1 ausführlich eingegangen. Als die hier angestellten Überlegungen illustrierendes Beispiel sei der „Bau“ eines Schneemannes angeführt, wozu einige Kugeln, Kegel und Zylinder durch Koordinaten bestimmender Punkte an geeigneten Positionen platziert werden müssen, so dass ihre Anordnung das gewünschte zusammengesetzte Objekt ergibt. Außerdem sind die Größenverhältnisse der einzelnen Bestandteile durch die Eingabe von Radien festzulegen.

## Zusammenfassung

Zusammenfassend vertrete ich folgende Thesen hinsichtlich der Entwicklung von Anschauungsvermögen und Raumvorstellung im Stoffgebiet Analytische Geometrie:

- Gravierende Verbesserungen des primären Raumvorstellungsvermögens sind in der Sekundarstufe II nicht zu erwarten. Die Entwicklung von Fähigkeiten des Koordinatisierens und räumlichen Strukturierens und im Zusammenhang damit einer „koordinatengebundenen Raumanschauung“ sollte jedoch ein wichtiges Ziel des Unterrichts sein, vgl. auch 1. auf S. 37.
- Unverzichtbare Grundvorstellungen, die Schüler in der analytischen Geometrie erlangen sollten, benötigen ein anschauliches Fundament, wie bereits aus der Beschreibung dieser Grundvorstellungen deutlich wird (siehe S. 37f.).
- Koordinatenbezogene Raumanschauung korrespondiert in elementarer Weise mit dem Denken in funktionalen Zusammenhängen, da Koordinaten gedanklich Positionen im Anschauungsraum zuzuordnen sind. Bei der Veranschaulichung durch Parameterdarstellungen beschriebener geometrischer Objekte tritt der Bezug zu funktionalen Betrachtungen besonders deutlich und in doppelter Hinsicht hervor, da Koordinaten (und somit Positionen) abhängig von Parameterwerten sind.
- Visualisierungshilfsmittel, insbesondere der Computer, können nur dann zum Aufbau anschaulicher Vorstellungen beitragen, wenn sie für die aktive geistige Auseinandersetzung mit Visualisierungen bzw. ihre Erstellung durch die Schüler selbst genutzt werden. Der Kombination von „Kopfgeometrie“ und Visualisierung mittels technischer Hilfsmittel kommt eine hohe Bedeutung zu. Computerunterstützte „was-wäre-wenn“-Betrachtungen, bei denen Schüler die Auswirkungen von Koordinaten- oder Koeffizientenänderungen auf die Lage von Punkten oder durch Gleichungen beschriebener geometrischer Objekte untersuchen, sind bedeutsam für die Entwicklung anschaulicher Vorstellungen und das Erkennen der genannten funktionalen Zusammenhänge. Ihr Nutzen bleibt jedoch begrenzt, wenn sie sich lediglich auf ein willkürliches „Ändern und Nachsehen“ beschränken. Vielmehr sollten Umkehrüberlegungen angestellt werden, wie Koordinaten bzw. Koeffizienten zu wählen sind, um eine gewünschte Lage zu erreichen.

Wie bereits ausgeführt wurde, kann die analytische Geometrie ihrem Wesen nach als visualisierende Disziplin angesehen werden; da sie umgekehrt geometrische Objekte und Relationen algebraisch beschreibt, stellt sie die Grundlagen der Generierung von Grafiken durch Computer bereit. Visualisierungen im Sinne geometrischer Interpretationen sind also keine „neue“ Arbeitsweise in der analytischen Geometrie, sondern dieser mathematischen Teildisziplin seit jeher wesensimmanent. Die Nutzung von Computervisualisierungen kann allerdings dazu beitragen, dieses Wesen hervortreten und den Bezug zwischen arithmetischen und algebraischen Beschreibungen sowie den zugehörigen geometrischen Objekten (Punktmengen) für die Schüler „erlebbar“ werden zu lassen. Häufig beklagtes schematisches Berechnen von Schnittmengen, Abständen und Winkeln ohne damit verbundene anschauliche Vorstellungen lässt sich dadurch reduzieren.

### 1.4.5 Vorschläge zur Überwindung der Formenarmut des Unterrichts in analytischer Geometrie – ein Überblick

Ein schwerwiegendes Defizit des im Unterricht hauptsächlich verfolgten Konzepts einer vektoriellen Geometrie des Raumes besteht darin, dass Schüler (vor allem in Grundkursen) oft nur lineare Objekte kennenlernen und untersuchen. Diese Formenarmut wurde vielfach, u. a. 1973 von FREUDENTHAL (siehe S. 19), 1979 von FÜHRER ([85]), 1987 von NICOL ([188]), 1992 von GIEDING ([91]) und 1993 von SCHMIDT in [218], beklagt.

Über die „Aufgabe des Geometrieunterrichts, seine Gegenstände als Vehikel der Welterschließung zu erarbeiten und anzuwenden“ schrieb SCHUPP 2000: „Am wenigsten tut er das in der S II. Lineare Gebilde und artifizielle Graphen sind nicht nur inner-, sondern auch außermathematisch entweder bloße Hilfen oder bedeutungsarm.“ Er kritisierte die „Überwucherung durch Beschreibungs- und Methodendruck“ und forderte, „die Methodendemonstration (die auf Dauer unweigerlich mit Objektverarmung verbunden ist) weit häufiger als bisher ab(zu)lösen, zumindest (zu) unterbrechen durch Objektexploration.“ ([234], S. 53–57)

Als wichtige, bereits von FREUDENTHAL herausgearbeitete, Ursache für die Formenarmut des Unterrichts ist die Unterordnung der Geometrie unter die Vektorrechnung und damit der betrachteten Inhalte unter die Methode anzusehen. „Es ist keine Frage, dass die heute in der S II vorherrschende Vektormethode ... zum Rückzug auf eine bloß lineare Geometrie geführt hat“ ([234], S. 56). Anhand des Beispiels eines hyperbolischen Paraboloids (konstruiert durch eine Schar von Verbindungsgeraden zwischen Punkten zweier nicht paralleler Diagonalen gegenüberliegender Seitenflächen eines Würfels) führte SCHUPP aus, dass das Studium interessanter geometrischer Objekte einer „Kombination vektorieller, traditionell analytischer, algebraischer, synthetischer und nicht zuletzt konstruktiver (sowohl direkt nachbildender als auch computergraphischer) Mittel“ bedarf ([234], S. 56). In der Tat lässt sich die Untersuchung von Kurven und Flächen nicht der Vektorrechnung unterordnen. Dies wird auch anhand der vielfältigen Vorschläge, die zur Behandlung von Kurven und Flächen im Unterricht unterbreitet wurden, deutlich. Da eine detaillierte Auswertung dieser Vorschläge den Rahmen dieser Arbeit erheblich überschreiten würde, sei im Folgenden eine kurze Zusammenfassung einiger Richtungen gegeben, in die Vorschläge zur Behandlung nichtlinearer geometrischer Objekte im Unterricht der analytischen Geometrie gingen bzw. gehen.

- Zur Behandlung der *Kegelschnitte* wurden nach ihrer weitgehenden Herausdrängung aus dem Mathematikunterricht mehrfach Vorschläge unterbreitet, u. a. von PROFKE ([205], [206]), SCHUPP ([235]) und TIETZE ([247], S. 224–238).<sup>88</sup> Diese

<sup>88</sup>Die Kegelschnitte finden sich auch in einigen neueren Schulbüchern (siehe Fußnote 45 auf S. 22), wenngleich sie z. B. in neueren Ausgaben des verbreiteten Lambacher Schweizer (siehe [8]) nicht mehr auftreten. In einigen Rahmenplänen sind sie als wahlobligatorische Gebiete enthalten, auch hierbei ist die Tendenz abnehmend, siehe Abschnitt 1.3. Jedoch deutet sich anhand der neuen Berliner und Hamburger Curricula für die 11. Jahrgangsstufe eine Tendenz zur elementaren analytisch-geometrischen Behandlung der Parabeln und teilweise auch Ellipsen und Hyperbeln an. Das Wort „Kegelschnitte“ wird dabei nicht erwähnt, da ihre Behandlung nur als ebene Kurven auf der Grundlage der Ortsdefinitionen bzw. Funktionsgleichungen vorgesehen ist.

Ansätze kommen größtenteils ohne Vektoren aus, wenngleich in [235] ein vektoriell-analytisches Verfahren zur Herleitung der Scheitelgleichung der Kegelschnitte als eine von mehreren Möglichkeiten erläutert wird. Bedeutung gewannen die Kegelschnitte auch im Zusammenhang mit elementargeometrischen Überlegungen zu Ortslinien, die durch die Einbeziehung dynamischer Geometriesoftware in den Geometrieunterricht der SI in den Blickpunkt rückten. Andere Eigenschaften von Kegelschnitten und Herangehensweisen ihrer Behandlung wurden ebenfalls in Unterrichtsvorschlägen aufgegriffen, so z. B. die Hüllkurveneigenschaft von GIEDING in [91]. Auch dieser Vorschlag geht von elementargeometrischen Konstruktionen aus und zu einer Koordinatenbeschreibung über, wobei der Einsatz eines CAS zur Veranschaulichung vorgesehen ist. Bereits 1989 untersuchte SCHUPP die Frage, inwieweit der Computer für die Konstruktion von Kegelschnitten sinnvoll nutzbar ist, und zeigte dafür Möglichkeiten unter Verwendung spezieller Turbo-Pascal-Programme auf. Er plädierte für eine Kombination aus traditionellen und computerunterstützten Methoden (vgl. [232]).

- M. WINTER beschrieb 1989 in [275] eine Unterrichtsreihe zur Behandlung von Schraubenlinien in Mathematik-Grundkursen. Ausgehend von der Betrachtung eines sich drehenden Flugzeugpropellers wurde dabei eine Parameterdarstellung der Schraubenlinie erarbeitet. Den Verlauf der Unterrichtsreihe, insbesondere das Interesse und die Mitarbeit der Schüler, schätzte er ausgesprochen positiv ein.
- In [233] entwickelte SCHUPP Vorschläge zur Behandlung von Kurven (Astroide, Rosette) als *Hüllkurven* einer gleitenden Strecke und *Ortslinien* der Lotfußpunkte vom Koordinatenursprung auf die Gleitstrecke. Auch hierbei erfolgt ein Übergang von elementargeometrischen Betrachtungen zu analytischen Beschreibungen.
- Einige Vorschläge, u. a. von MEYER ([183]) und WUNDERLING ([279]), zielten darauf ab, ausgehend von *Funktionsgraphen* durch Verallgemeinerung und den Übergang zu *Parameterdarstellungen* bestimmte Klassen von *Kurven* (u. a. kubische Kurven) mit den Mitteln der analytischen Geometrie und der Analysis zu untersuchen. MEYER ging in [183] über die Behandlung ebener Kurven hinaus und betrachtete einige Raumkurven, wobei er mithilfe von CAS generierte Visualisierungen einbezog. KROLL schlug in [141] die Behandlung ebener Kurven im Raum auf der Grundlage von Funktionsgleichungen der Form  $t = f(s)$  mit den beiden Parametern  $s$  und  $t$  einer Ebene vor. Die Funktionen und ihre Graphen untersuchte er u. a. mit Mitteln der Analysis. Vorschläge zur Behandlung von Kurven im Unterricht der analytischen Geometrie finden sich auch bei TIETZE ([247], S. 239–260), der u. a. Objektstudien zum Kreis und zu Spiralen beschrieb.
- Auf die Behandlung von *Flächen* gingen SCHROTH und TIETZE in [247], S. 261–277 ein. Sie bezogen sich insbesondere auf Graphen von Funktionen zweier Variablen,<sup>89</sup> Flächen 2. Ordnung, Rotationsflächen und Regelflächen, wobei sie vor

---

<sup>89</sup>Ähnliche Vorschläge von FLACHSMEYER und WEIGAND zur Untersuchung von Funktionen zweier Variablen mit besonderem Augenmerk auf die Nutzung des Computers finden sich in [264].

allem experimentelle Untersuchungen ins Auge fassten. In [181] beschrieb MEYER, ausgehend von parametrisierten Kurvenscharen, die Behandlung spezieller kubischer Flächen im Raum mit koordinatengeometrischen und an einigen Stellen auch vektoriellen Methoden. In [184] ging derselbe Autor auf die Untersuchung von Kurven und Flächen (u. a. Sattelflächen) ein und verwendete in stärkerem Maße vektorielle Vorgehensweisen (hauptsächlich für die Beschreibung erzeugender Geraden sowie von Tangenten und Normalen). Die Nutzung des Computers zu Visualisierungs- und Experimentierzwecken ist eine wichtige Voraussetzung für die Umsetzung der genannten Vorschläge zur Untersuchung von Flächen.

- Zunehmend interessant wurden in den letzten fünf Jahren *Freiformkurven* (und ansatzweise -flächen), insbesondere *Bézierkurven*, die elementargeometrische Erzeugungsmethoden, analytische Beschreibungen sowie algorithmische Aspekte miteinander in Verbindung bringen. Unterrichtsvorschläge zu Bézierkurven unterbreiteten KLEIFELD in [131] und [132] sowie MEYER in [179] und [180] (in diesem Beitrag sowie in [184] ging er auch ansatzweise auf Bézierflächen ein). ROTH entwickelte in [213] ein Konzept für die Behandlung von Bézierkurven in Grundkursen; von DZUNG WONG erschien 2003 ein Schülermaterial zu Bézierkurven ([53]). Da Bézier- und Splinekurven sowie -flächen eine hohe Bedeutung für die Modellierung von Formen in der Computergrafik haben und ihre Entwicklung sogar für diesen Zweck erfolgte, wird darauf in der vorliegenden Arbeit noch näher eingegangen (siehe Abschnitte 2.3.6 und 2.3.8).

Diese (mit Sicherheit unvollständige) Aufzählung zeigt, dass es eine Reihe von Vorschlägen gab und gibt, den Unterricht in analytischer Geometrie vielseitiger, „formenreicher“ und damit attraktiver zu gestalten, die auch erfolgreich umgesetzt wurden. Bemerkenswert erscheint u. a., dass ROTH in dem genannten Beitrag ein m. E. überzeugendes Konzept einer das selbstständige Lernen der Schüler herausfordernden, anwendungsorientierten Behandlung interessanter Kurven in Grundkursen skizzierte. Auch die erwähnten Ausführungen von WINTER zur Behandlung der Schraubenlinie verdeutlichen, dass in Grundkursen ein anspruchsvoller Mathematikunterricht bei einer hohen Beteiligung der Schüler möglich ist. Viele der anderen aufgeführten Vorschläge zielen eher auf Leistungskurse mit interessierten und leistungsstarken Schülern.

Den skizzierten Ansätzen ist gemeinsam, dass die verwendeten *Methoden* (elementar- und koordinatengeometrische, vektorielle und zum Teil auch algorithmische Methoden sowie Verfahren der Analysis) nach ihrer Zweckmäßigkeit für die Untersuchung der betrachteten *Objekte* ausgewählt wurden, was der eingangs dieses Abschnitts zitierten Forderung von SCHUPP entspricht. Diese Forderung erscheint auch insofern bedeutsam, als die Leistungsfähigkeit von Abstraktionen (wie z. B. des Vektorbegriffs), Beschreibungen (wie impliziten Gleichungen und Parameterdarstellungen) sowie Verfahren (wie der Zurückführung von Schnittmengen auf Lösungsmengen von Gleichungssystemen) für die Schüler nicht genügend nachvollziehbar wird, wenn sie diese nur an wenigen und zudem recht uninteressanten Objekten erfahren bzw. anwenden.

Für Untersuchungen geometrischer Objekte, bei denen den zu betrachtenden Objekten Priorität gegenüber den Arten der mathematischen Beschreibung und den Methoden der Untersuchung eingeräumt wird, findet sich häufig der Terminus *Objektstudien*. Bereits 1979 beschrieb FÜHRER in [85] diesen Ansatz anhand raumgeometrischer Beispiele konvexer Mengen und beschrieb Unterrichtserfahrungen u. a. bei der Untersuchung als Vektorsummen „konstruierter“ geometrischer Körper (z. B. Tori). TIETZE zeigte in [247] eine Reihe von Möglichkeiten einer am Ansatz der Objektstudien orientierten Behandlung geometrischer Körper, Kurven und Flächen auf.<sup>90</sup> WITTMANN beschrieb in [73] Objektstudien an Tetraedern und Oktaedern als vor allem für Grundkurse geeignete Möglichkeit des Einstiegs in die räumliche analytische Geometrie. Dabei wird von anschaulichen elementargeometrischen Überlegungen ausgegangen, wobei sich jedoch recht bald eine Koordinatisierung als notwendig erweist und schließlich auch Grenzen der klassischen Koordinatengeometrie deutlich werden:

„Werden im Zuge von Objektstudien die Möglichkeiten einer vektorfreien Koordinatengeometrie „ausgereizt“, besteht ein natürlicher Anlass, neue Methoden einzuführen.“ ([73], S. 97)

### Zusammenfassung und Ausblick

Der Unterricht in analytischer Geometrie bedarf *einer größeren Vielfalt zu untersuchender interessanter geometrischer Objekte* und einer *flexibleren Herangehensweise bei der Auswahl von Beschreibungen und Methoden*. Heuristische und experimentelle Vorgehensweisen, wie sie durch den Einsatz des Computers z. T. erst ermöglicht werden, können sich dabei als sehr nützlich erweisen. Der Computer eröffnet zudem Chancen zur Veranschaulichung interessanter räumlicher Kurven und Flächen, die ohne Visualisierungshilfsmittel in der Schule nur schwerlich behandelt werden könnten.

Computergrafik ist somit ein Hilfsmittel für eine anspruchsvolle analytische Raumgeometrie. Sie kann aber auch die Beschäftigung mit geometrischen Objekten provozieren. So erfordert die Erzeugung von Animationen (ein erfahrungsgemäß für Schüler besonders interessantes Thema) die parameterabhängige Beschreibung von Bewegungsbahnen und motiviert somit die Behandlung zumindest recht einfacher Kurven wie z. B. Kreise, Spiralen und Schraubenlinien; auf diesen Aspekt wird in der vorliegenden Arbeit in den Abschnitten 2.6 und 4.3 ausführlicher eingegangen.

---

<sup>90</sup>Eine Fülle interessanter Ansätze für Objektstudien geometrischer Objekte (insbesondere Kurven und Flächen) mit vielen Anwendungsbezügen enthält auch [93], worin zwar die beschriebenen Kurven und Flächen hauptsächlich phänomenologisch betrachtet, Vertiefungsmöglichkeiten jedoch an vielen Stellen deutlich werden; so liefern allein z. B. die Abschnitte zu Schraubenlinien und Schraub- sowie Spiralfächen (S. 210–224) eine Vielzahl reichhaltiger Anknüpfungspunkte.

### 1.4.6 Anwendungsbezug und mathematische Modellierung

Zur Berücksichtigung von Anwendungsbezügen im Unterricht der analytischen Geometrie und linearen Algebra wurden im Verlauf der letzten Jahrzehnte vielfältige Überlegungen angestellt und Vorschläge unterbreitet, von denen viele auf die Behandlung einer anwendungsorientierten linearen Algebra abzielen.<sup>91</sup>

Von Bedeutung sind Anwendungsbezüge im Mathematikunterricht u. a. hinsichtlich daran orientierter inhaltlicher Zugänge zu mathematischen Themengebieten sowie in Bezug auf die Motivierung. So soll bei Schülern Interesse für die Beschäftigung mit mathematischen Inhalten geweckt werden, indem ihnen deren Praxisrelevanz deutlich wird. Allerdings stellt sich häufig heraus, dass Schüler sehr deutlich „Pseudoanwendungsaufgaben“ erkennen, bei denen lediglich Aufgaben in (zumeist isolierte) reale Sachverhalte „eingekleidet“ sind. Bestenfalls kurzzeitig motivierend ist es auch, Anwendungsmöglichkeiten mathematischer Inhalte bei deren Behandlung lediglich zu nennen. Vielmehr kommt es darauf an, dass die Schüler selbst mathematische Modellbildungen erarbeiten und deren Nutzen in für sie relevanten Anwendungen erleben.

Didaktische Konzepte für die komplexere Berücksichtigung von Anwendungen und Real-situationen im Mathematikunterricht wurden in den vergangenen Jahren häufig unter Aspekten *mathematischer Modellierungsprozesse* untersucht. Im Unterschied zu Anwendungsaufgaben, bei denen mathematische Unterrichtsinhalte zur Lösung entsprechend ausgewählter (und meist bereits „passgerecht“ aufbereiteter) realitätsbezogener Aufgaben genutzt werden, umfasst die mathematische Modellierung auch

- Überlegungen zur Vereinfachung und Strukturierung realer Situationen durch Beschränkung auf die wesentlichen Aspekte (Idealisierung),
- die Mathematisierung von Strukturen und Zusammenhängen sowie
- die Interpretation der am mathematischen Modell erarbeiteten Lösungen auf die reale Ausgangssituation (siehe u. a. [110], S. 1–5).

Einen Vorschlag für die mathematische Modellierung von Realsituationen im Unterricht der analytischen Geometrie unterbreitete MAASS in [170]. Sie griff Fragen der Flugsicherung als Anwendung von und Motivierung für Schnittbestimmungen zwischen Geraden und Ebenen auf (siehe auch [247], S. 169f.). ABEL und HAUBROCK behandelten unter Bezug auf die Wirkungsweise von Navigationssystemen (GPS) Schnittkreise und -ebenen von Kugeln.<sup>92</sup> Bereits in Abschnitt 1.4.5 wurden einige Ansätze der mathematischen Modellierung von Realsituationen aufgeführt, die zu interessanten Kurven wie Schraubenlinien und Bézierkurven führen.

---

<sup>91</sup>Siehe dazu Abschnitt 1.2.4; im Rahmen der vorliegenden Arbeit interessieren allerdings vor allem geometrisch orientierte Anwendungen, weshalb Vorschläge zur Behandlung nichtgeometrischer Anwendungen der linearen Algebra nicht ausführlicher erörtert werden.

<sup>92</sup>Die von ABEL behandelte Anwendung (siehe [1]) erfordert für eine realistische Modellierung etwas umfangreichere Voraussetzungen, nicht nur aus der analytischen Geometrie, sondern auch aus der Analysis bzw. Numerik (Newton-Verfahren), und führt somit zu stoffgebietsübergreifenden Betrachtungen. HAUBROCK stellte in [107] die Geometrie der Kugeloberfläche bei der Behandlung von GPS in den Mittelpunkt. Für die Simulation realer Berechnungen von GPS ist der Einsatz eines CAS sinnvoll.

Häufig sind bei der Modellierung realer Sachverhalte Approximations- und Interpolationsverfahren von Bedeutung. Werden Spline-Funktionen<sup>93</sup> zur Interpolation genutzt, so ergeben sich anhand von Anwendungen Bezüge zwischen der Analysis, numerischen Verfahren und geometrischer Modellierung, wie KLINCSIK in [135] anhand von Beispielen und Unterrichtsvorschlägen ausführte.

In der dreidimensionalen Computergrafik treten „Modellierungen“ in mehreren Zusammenhängen auf:

- Bei der *geometrischen Modellierung* werden Formen (Flächen bzw. Körper) mithilfe einer Beschreibung durch Koordinaten bzw. Gleichungen generiert. Es handelt sich hierbei also um Modellierung im Sinne von Konstruktion oder „Modellbau“.<sup>94</sup>
- Die fotorealistische 3D-Computergrafik simuliert Prinzipien der Lichtausbreitung in der Realität anhand von *Beleuchtungsmodellen*. Die mathematische Beschreibung von Beleuchtung auf der Grundlage optischer Gesetze der Lichtreflexion und -brechung ist eine Grundlage von 3D-Grafikprogrammen.<sup>95</sup>

Der Begriff „Modellierung“ erhält bei der Behandlung der 3D-Computergrafik im Unterricht noch in anderer Hinsicht eine interessante Doppelbedeutung. Zum einen liegen ihr, wie schon erwähnt wurde, Modelle der Lichtausbreitung und der Fotografie zugrunde. Zum anderen ist 3D-Computergrafik selbst bereits ein Bestandteil der „Realität“ (insbesondere für Jugendliche) geworden.<sup>96</sup> Wenn mathematische Grundlagen der Computergrafik und damit die Funktionsweise z. B. von Computerspielen untersucht werden, so erfolgt damit eine „Modellierung“ von zwei Ebenen der Realität: der physikalischen und der „virtuellen“ Realität, die für viele (insbesondere spielbegeisterte) Jugendliche ebenso Teil ihrer Lebensumgebung ist. Verfolgt man diesen Gedanken weiter, so beinhaltet die mathematische Beschreibung dreidimensionaler Szenen durch die Schüler die „Schaffung von Realität“ im Sinne der Konstruktion virtueller Welten.

---

<sup>93</sup>Auf Spline-Funktionen und -kurven, die (wie auch Bézierkurven) häufig für Zwecke der geometrischen Modellierung in der Computergrafik zum Einsatz kommen, wird in Abschnitt 2.3.6 der vorliegenden Arbeit noch näher eingegangen.

<sup>94</sup>Dabei muss es sich nicht einmal um Modellierung im Sinne von „Nachbau“ real vorhandener Objekte handeln. Auch die Schaffung von Objekten im Sinne der Modellierung geometrischer Formen, die nur in der Vorstellung existieren oder an die bestimmte Anforderungen gestellt werden, ist möglich und hat im computergestützten Entwurf (CAD) eine hohe Bedeutung. Die in Abschnitt 1.4.5 erwähnten Vorschläge zur Behandlung von Bézierkurven verfolgen diesen Ansatz der Modellierung.

<sup>95</sup>Sowohl Aspekte der geometrischen Modellierung als auch Beleuchtungsmodelle in der Computergrafik werden im 2. Kapitel dieser Arbeit beschrieben und bilden Schwerpunkte der in Kapitel 4 unterbreiteten Unterrichtsvorschläge. Dazu gehört die Motivierung und Erarbeitung des Skalarproduktes und von Normalenvektoren im Kontext der Beleuchtung von Körpern sowie eine Rückübertragung der mathematischen Überlegungen mithilfe von Computerexperimenten.

<sup>96</sup>So stellte SCHUPP bereits 1996 fest, dass der Computer ein Stück „Lebenswirklichkeit“ geworden ist ([113], S. 199). Diese Tendenz hat sich seither erheblich verstärkt und trifft auf die Computergrafik in starkem Maße zu, die Jugendlichen in Videoclips, Filmen, Zeitschriften und vielen anderen Zusammenhängen, besonders intensiv aber in Computerspielen begegnet.



### 1.4.7 Vernetzungen zwischen analytischer Geometrie und Elementargeometrie sowie Analysis

Stoffgebietsübergreifende Betrachtungen und Anwendungen können dazu beitragen, dass Schüler die Mathematik nicht als Summe zusammenhangloser Teilgebiete kennenlernen, sondern einen Einblick in ihre Leistungsfähigkeit als Gesamtdisziplin erhalten. *Vertikale Vernetzungen*, das Anknüpfen an Lerninhalte früherer Klassenstufen, sind im Sinne des BRUNERSchen Spiralprinzips auch für das Erreichen eines vertieften Verständnisses mathematischer Inhalte durch ihre Behandlung auf verschiedenen Abstraktionsstufen von Bedeutung. Für das Stoffgebiet Analytische Geometrie bieten sich vor allem Bezüge zu folgenden Themengebieten des Mathematikunterrichts der SI an:

- *lineare Gleichungen und Gleichungssysteme*, die meist in Klasse 9 für LGS mit zwei Variablen und zwei Gleichungen rechnerisch und zeichnerisch gelöst werden,
- *elementare Funktionen und ihre Eigenschaften* (falls in der analytischen Geometrie nichtlineare geometrische Objekte behandelt werden),
- Elementargeometrie der Ebene und des Raumes.

Bezüge zum Lösen linearer Gleichungssysteme in der SI werden bei der Behandlung von LGS im Rahmen des Stoffgebietes Lineare Algebra / Analytische Geometrie beinahe zwangsläufig hergestellt. Die Anknüpfung an die elementaren Funktionen kann im Zusammenhang mit der in Abschnitt 1.4.5 diskutierten Aufnahme nichtlinearer geometrischer Objekte in den Unterricht erfolgen. Da elementare Funktionen auch Untersuchungsgegenstand des Unterrichts in Analysis sind, treten hierbei horizontale Vernetzungen in den Vordergrund.

#### Analytische Geometrie und Elementargeometrie

Die ergiebigsten Bezüge der analytischen Geometrie zum Mathematikunterricht der Sekundarstufe I bestehen m. E. zur Elementargeometrie, insbesondere des Raumes.<sup>97</sup> Auch ein Anknüpfen an die ebene Geometrie und ihre Weiterführung mit analytischen Methoden bei gleichzeitiger Ausweitung auf den Raum erscheint angebracht. So ist es als „Zwischenstufe“ zwischen elementargeometrischen Definitionen von Figuren und Körpern sowie ihrer Beschreibung durch Gleichungen bzw. Parameterdarstellungen sinnvoll, Körper durch die Koordinaten charakteristischer Punkte und charakteristische Größen darzustellen.<sup>98</sup> Hierbei handelt es sich auch um eine wichtige Vorgehensweise der geometrischen Modellierung in der 3D-Computergrafik.

---

<sup>97</sup>Die Raumgeometrie ist in der SI leider unterrepräsentiert; so formulierte FÜHRER, der „*allgemeinbildenden ... Schule fehlt eine ... Geometrie, die den Raum von Objekten und Operationen, Koordinations- und Bewegungserfahrungen her konstruktiv aufbaut*“ ([86], S. 10). Dennoch erscheint ein Aufgreifen und Vertiefen der recht spärlichen raumgeometrischen Inhalte des MU der SI im Stoffgebiet Analytische Geometrie wünschenswert. Die Einbeziehung von Elementen der 3D-Computergrafik ermöglicht dabei konstruktive Vorgehensweisen auf der Grundlage von Koordinatenbeschreibungen.

<sup>98</sup>So kann die elementargeometrische Definition eines Kreises als Menge aller Punkte einer Ebene, die von einem gegebenen Punkt denselben Abstand haben, leicht auf entsprechende Punktmengen im Raum, also Kugeln, übertragen werden. Diese werden dementsprechend durch die Koordinaten

Weitere Bezüge zwischen der Elementargeometrie der S I und der analytischen Geometrie liegen u. a. im Beweisen elementargeometrischer Sätze mithilfe vektorieller Methoden und in der analytischen Beschreibung geometrischer Abbildungen (Bewegungen, zentrische Streckungen und Projektionen).<sup>99</sup> Vorschläge, elementargeometrische sowie analytisch-geometrische Überlegungen und Methoden bei der Behandlung von Kurven (von Kegelschnitten bis zu Bézierkurven) sowie anhand von Objektstudien platonischer Körper miteinander zu verbinden, wurden bereits in Abschnitt 1.4.5 skizziert.

Vernetzungen zwischen der Elementargeometrie und der analytischen Geometrie müssen sich nicht ausschließlich auf Unterrichtsinhalte beziehen, auch das Aufgreifen für den Geometrieunterricht der S I spezifischer *Betrachtungs- und Arbeitsweisen* kann dazu beitragen. Dazu zählen neben anschaulichen und lokal-deduktiven Aspekten operative, von praktischen, zweckgerichteten Handlungen ausgehende Herangehensweisen. Diese wurden von BENDER und SCHREIBER untersucht und begründen ein „Prinzip der operativen Begriffsbildung“ ([14], S. 26). Geometrische Vorstellungen entstehen danach in starkem Maße durch „Konstruktionen in der unmittelbaren handwerklich-technischen Praxis“ ([95], S. 171). Im Stoffgebiet Analytische Geometrie besteht m. E. die Möglichkeit, an dieses Prinzip anzuknüpfen, wobei „Konstruktionen“ natürlich mit analytischen Methoden, d. h. mittels Koordinaten- oder Gleichungsdarstellungen erfolgen sollten. Die Nutzung von Computergrafik-Software bietet die Möglichkeit, derartige Konstruktionen (mit sichtbaren Ergebnissen) durchzuführen; es besteht dabei ein enger Zusammenhang zu geometrischen Modellierungen (vgl. Abschnitt 1.4.6).

### Vernetzungen zwischen analytischer Geometrie und Analysis

Hinsichtlich horizontaler Vernetzungen zwischen der analytischen Geometrie und den beiden anderen „großen“ Themengebieten des Mathematikunterrichts der S II sind vor allem Bezüge zur Analysis bedeutsam.<sup>100</sup> Dabei liegen folgende Ansätze nahe:

- Einbeziehung elementarer Funktionen in den Unterricht der analytischen Geometrie und ihre Nutzung für die Beschreibung von Kurven in der Ebene oder (durch Erweiterung auf Funktionen zweier Variablen) von Flächen im Raum;<sup>101</sup>

---

des Mittelpunktes und den Radius beschrieben. Fragen der geometrischen Modellierung können als Motivierung genutzt werden, eine Gleichung für die Punkte einer Kugel aufzustellen. Auch hierbei ist ein Aufgreifen von Unterrichtsinhalten der S I (vor allem des Satzes des Pythagoras) sinnvoll. Auf derartige Betrachtungen als Einstieg in die analytische Geometrie wird in dieser Arbeit noch ausführlich eingegangen, wobei die Modellierung dreidimensionaler Szenen der Motivierung dient.

<sup>99</sup>Das „exemplarische Ausführen von Abbildungen“ innerhalb der analytischen Geometrie unter Nutzung des Computers wurde bereits 1993 von G. SCHMIDT gefordert (vgl. [218], S. 27). Vorschläge, die auf die Beschreibung von Abbildungen durch Matrizen und ihre Ausführung mithilfe des Computers abzielen, kommen in den Abschnitten 1.5 und 3.1 kurz zur Sprache.

<sup>100</sup>Es existieren auch Anknüpfungspunkte zwischen Geometrie und Stochastik, deren Behandlung aber weniger nahe liegend erscheint als Bezüge zwischen analytischer Geometrie und Analysis.

<sup>101</sup>Dabei erfolgt zwar keine Verwendung von Elementen der Infinitesimalrechnung, es werden jedoch Bezüge durch die Einbeziehung elementarer Funktionen sichtbar. Eine derartige Behandlung ebener Kurven sieht der neue Berliner Rahmenplan für die Klassenstufe 11 vor, vgl. S. 29.

- Untersuchung von Kurven, die auch in Parameterdarstellung gegeben sein können, mithilfe der Differenzialrechnung – dazu ist ein erweitertes Verständnis von Ableitungen (von Vektoren bzw. Koordinatenpaaren oder -tripeln nach Parametern) erforderlich;<sup>102</sup>
- Anwendung von Methoden der Analysis auf räumliche Objekte.

Aufgrund der Ausrichtung des Unterrichts auf die analytische Geometrie des Raumes erscheint natürlich die letztgenannte Variante besonders erstrebenswert, sie wirft jedoch mindestens zwei Probleme auf:

- Im gegenwärtigen Unterricht der analytischen Geometrie werden kaum nichtlineare geometrische Objekte behandelt.
- Die Anwendung der Differenzialrechnung auf räumliche Objekte (vor allem Flächen) geht über den Unterrichtsstoff in Analysis, der i. Allg. ausschließlich Ableitungen von Funktionen einer Variablen beinhaltet, deutlich hinaus.

Sollen Flächen des Raumes (z. B. Kugelflächen, die in vielen Rahmenplänen auftreten) mit Mitteln der Analysis untersucht werden, so ist ein intuitiver Zugang zu partiellen Ableitungen notwendig. Sinnvoll dürfte in den meisten Fällen eine Beschränkung auf Flächen sein, die durch Funktionsgleichungen zweier Variablen darstellbar sind, wie z. B. Halbkugeln. Eine anschauliche Möglichkeit, partielle Ableitungen einzuführen, besteht in der Betrachtung von Schnittkurven der Fläche mit zu Koordinatenebenen parallelen Ebenen und die Ableitung der (durch das Setzen einer Variablen als Konstante entstehenden) Funktionsgleichungen dieser Kurven.<sup>103</sup>

Einige Möglichkeiten zur Herstellung von Querverbindungen zwischen der Analysis (oder zumindest den elementaren Funktionen) und der analytischen Geometrie, die auf die Nutzung von 3D-Computervisualisierungen aufbauen bzw. im Zusammenhang mit der Funktionsweise der 3D-Computergrafik stehen, werden in der vorliegenden Arbeit beschrieben. Die (sowohl gedankliche als auch visuelle) Betrachtung von Schnittkurven mit den Koordinatenebenen oder dazu parallelen Ebenen wird sich dabei als eine zentrale Arbeitsweise herausstellen, welche häufig die Zurückführung dreidimensionaler auf zweidimensionale Probleme ermöglicht.

---

<sup>102</sup>Einige in diese Richtung gehende Vorschläge wurden bereits in Abschnitt 1.4.5 skizziert. Ihnen ist gemeinsam, dass sie sehr interessante Untersuchungen ermöglichen, aber auf einem hohen Niveau angesiedelt sind. Sie eignen sich deshalb vor allem für den Unterricht mit überdurchschnittlich motivierten und begabten Schülern in Leistungskursen.

<sup>103</sup>In [66] wurde die Herleitung der Gleichung der Tangentialebenen an eine Kugel (genauer: Halbkugel) durch Ableitung der Gleichungen der Schnittkurven der Halbkugel mit zu Koordinatenebenen parallelen Ebenen beschrieben. Dieser Unterrichtsvorschlag ließ sich in einem leistungsstarken und interessierten Leistungskurs erfolgreich umsetzen, wobei das Aufgreifen von Mitteln der Analysis im Stoffgebiet Analytische Geometrie die Schüler regelrecht überraschte und Interesse hervorrief. Allerdings erfordert das beschriebene Vorgehen anspruchsvolle Überlegungen, welche erfahrungsgemäß die Mehrzahl der Schüler in Grundkursen überfordern. Hier erscheint die Herstellung von Querverbindungen zwischen Analysis und analytischer Geometrie anhand ebener Kurven realistischer – ein geeignetes Beispiel dafür ist die Bestimmung der Tangentengleichung an Halbkreise (siehe auch die auf Grundlage von [66] entstandenen Schulbuchseiten [223], S. 197 und [224], S. 217f.).

### 1.4.8 Unterrichtskultur und Unterrichtsmethoden

Während in den vorangegangenen Abschnitten hauptsächlich auf Vorschläge zur Überwindung von Problemen des Unterrichts in analytischer Geometrie anhand der zu betrachtenden Unterrichtsinhalte eingegangen wurde, sollen nun einige Aspekte der verwendeten Unterrichtsformen diskutiert werden, die allerdings immer auch von den behandelten Inhalten ab- bzw. mit diesen zusammenhängen.

Vielfach folgt die Gestaltung des Mathematikunterrichts – nicht nur im Stoffgebiet Analytische Geometrie, aber hier vielleicht besonders stark ausgeprägt – der von LENNÉ als „Aufgabendidaktik der traditionellen Mathematik“ bezeichneten Vorgehensweise:

*„In der Aufgabendidaktik werden – zugespitzt formuliert – jeweils bestimmte Kenntnisse, Operationen und Methoden vom Lehrer vorgetragen. Noch immer ist auch der Arbeitsunterricht<sup>104</sup> – schon aus Zeitmangel – oft nur ein suggestiv gesteuerter Lehrervortrag aus dem Munde der Schüler. Sodann werden vom Lehrer beziehungsweise Lehrbuch Aufgaben gestellt. Diese Aufgaben werden vom Schüler gelöst. Die Resultate werden vom Lehrer kontrolliert.“*

([154], S. 51)

Sicherlich hat fragend-entwickelnder Unterricht nach wie vor eine Daseinsberechtigung und auch Lehrervorträge können im Unterricht, speziell in der S II, die auf ein Hochschulstudium vorbereiten soll, sehr sinnvoll eingesetzt werden. Jedoch ist es inzwischen Konsens in der Mathematikdidaktik, dass der Mathematikunterricht auch weite Phasen selbsttätigen Lernens umfassen sollte.<sup>105</sup>

Selbstständiges Erarbeiten mathematischer Inhalte durch die Schüler ist nicht auf der Ebene von Abstraktionen möglich, für welche diese keine hinreichend breite Basis von Beispielen und Repräsentanten zur Verfügung haben – eine selbsttätige Erarbeitung des Vektor- oder gar des Vektorraumbegriffs durch die Schüler zu Beginn des Stoffgebietes Analytische Geometrie dürfte z. B. unmöglich sein. Auch die von SCHUPP beklagte „Dominanz der Methoden“ (vgl. S. 45) ist eng mit Kalkülorientierung und „Aufgabendidaktik“ verbunden. Selbstständige Wissens- und auch Methodenaneignung wird demgegenüber durch die Untersuchung konkreter Objekte und durch die Modellierung realer Situationen<sup>106</sup> begünstigt. Auch Vernetzungen können durch die Möglichkeit des Anknüpfens an Vorwissen zu selbstständiger Erkenntnisfindung beitragen.

<sup>104</sup>LENNÉ verwendet den Begriff „Arbeitsunterricht“ in diesem Zusammenhang im Sinne der heute gebräuchlichen Bezeichnung „fragend-entwickelnder Unterricht“.

<sup>105</sup>So wurde in der bereits auf S. 33 erwähnten Expertise zum Mathematikunterricht in der gymnasialen Oberstufe konstatiert, dass nur eine „offene, prozessorientierte Unterrichtskultur der Bedeutung der Heuristik für das Lernen von Mathematik gerecht werden kann. ... Da Lernen konstruktiv ist, spielt die Selbsttätigkeit der Schülerinnen und Schüler eine wichtige Rolle ...“ ([29], S. 76).

<sup>106</sup>HUSSMANN entwickelte in [120] Vorschläge, anhand „intentionaler Probleme“ (die im Spannungsfeld zwischen Unterrichtsstoff und Erfahrungswelt der Schüler stehen, den Zugang zu einem Themengebiet eröffnen und weit in das Thema tragen sollen, [120], S. 6 und 23) selbsttätiges Lernen anzuregen. Hinsichtlich der analytischen Geometrie diskutierte er drei derartige intentionale Probleme (Flugsicherheit, Schiffsrouten und Zusammenhänge zwischen Qualitätskriterien an Internetseiten). HUSSMANNs Konzepte haben Gemeinsamkeiten mit Vorschlägen zum mathematischen Modellieren (vgl. 1.4.6), seine Ausgangspunkte sind jedoch weniger inhaltlichen denn lerntheoretischen Ursprungs.

Es würde über den Rahmen dieser Arbeit hinausgehen, allgemein Konzepte schülerzentrierten Unterrichts und selbstständigen sowie kooperativen Lernens zu erörtern. Vielmehr werden in den Kapiteln 3, 4 und 5 entsprechende Möglichkeiten diskutiert, die sich im Zusammenhang mit der Einbeziehung von Elementen der 3D-Computergrafik in das Stoffgebiet Analytische Geometrie ergeben.

Generell lässt sich der Computer an vielen Stellen im Mathematikunterricht als wichtiges Hilfsmittel für experimentelles und heuristisches Arbeiten nutzen. Von den Schülern als lohnenswert angesehene anwendungspraktische Ziele können dabei zu der Notwendigkeit führen, mathematische Inhalte zu erarbeiten. Hierdurch bieten sich Ansatzpunkte für den Lehrer, als „Berater“ in Erscheinung zu treten und Fragen der Schüler aufzugreifen. Allerdings ist festzuhalten, dass nicht alle sinnvollen und notwendigen Unterrichtsinhalte auf diese Weise zu behandeln sind. Instruktionen durch den Lehrer werden an einer Reihe von Stellen, insbesondere bei der Erarbeitung von abstrahierenden Begriffen sowie von Verfahren, nach wie vor erforderlich sein. Trotz der eingangs dieses Abschnitts geäußerten Kritik an einer „Aufgabendidaktik“ als dominierendem Unterrichtsprinzip haben auch Übungsphasen, in denen Schüler mehrere Aufgaben desselben Typs lösen, eine Bedeutung für die Festigung wichtiger Verfahren. So sehr Phasen entdeckenden Lernens sinnvoll und wünschenswert sind, so unakzeptabel ist m. E. die Verabsolutierung dieses Ansatzes und die mitunter anzutreffende Ablehnung jeder lehrerzentrierten Vorgehensweise.

TIETZE fasste die vielfältigen Vorschläge und Diskussionen um Herangehensweisen und Ziele des Mathematikunterrichts in der Sekundarstufe II sowie entsprechender Veränderungen der Unterrichtskultur zu folgenden, m. E. sehr ausgewogenen, Forderungen zusammen ([247], S. 151):

1. *„der Mathematikunterricht soll allgemeinbildend und wissenschaftspropädeutisch sein und damit zu einer allgemeinen Studierfähigkeit beitragen;*
2. *er vermittelt ein angemessenes Bild von Mathematik, das insbesondere auch mathematische Modellbildung sowie den algorithmusorientierten und experimentellen Aspekt von Mathematik umfasst; er soll problem- und anwendungsorientiert sein, aber auch Sicherheit in Basis-Routinen und ein angemessen vernetztes Wissen vermitteln;*
3. *er ist inhaltlich und methodisch an den fundamentalen Ideen, die dem Gebiet zugrunde liegen, zu orientieren;*
4. *er soll den Kenntnissen, Fähigkeiten und Motiven der unterrichteten Schülergruppe angemessen sein; er soll konstruktives und experimentelles Arbeiten ermöglichen; spielerische Elemente, die Kreativität fördern können, sind wünschenswert;*
5. *die Inhalte sollen vom Lehrer überzeugend vermittelt werden können; eigene mathematische Neugier und Interesse können den Unterricht vor Erstarrung und Routine bewahren; das Experimentieren mit einem CAS oder einem dynamischen Geometrie-programm kann auch für den Lehrer spannend sein;*
6. *die vorangegangenen Forderungen verlangen ein breites Spektrum von Unterrichtsverfahren; neben den üblichen fragend-entwickelnden Unterricht treten Phasen eines entdeckenlassenden Unterrichts und (kleinere) Projekte; auch kurze Phasen eines expositorischen Unterrichts sind sinnvoll.“*

## 1.5 Zur Computernutzung im Unterricht der analytischen Geometrie

### 1.5.1 Didaktische Aspekte des Computereinsatzes im Mathematikunterricht

Fragen des Computereinsatzes im Mathematikunterricht nahmen in der didaktischen Diskussion der vergangenen drei Jahrzehnte eine wichtige Stellung ein. Dabei kristallisierten sich einige Hauptaspekte heraus, hinsichtlich derer die Verwendung des Computers zu einer Weiterentwicklung des Mathematikunterrichts betragen kann.

- (1) Die Nutzung des Computers ermöglicht *heuristisches und experimentelles Arbeiten* in anspruchsvollen Kontexten, für die vorher keine geeigneten „Experimentierhilfsmittel“ zur Verfügung standen. Schüler können Gesetzmäßigkeiten und Zusammenhänge selbst herausfinden.<sup>107</sup> Die Verwendung des Computers als heuristisches und experimentelles Hilfsmittel ist auch im Zusammenhang mit der Öffnung von Aufgaben interessant (siehe z. B. [270]).
- (2) Durch den Einsatz des Computers ergeben sich erweiterte Möglichkeiten für *kreatives Arbeiten*<sup>108</sup> im Sinne der Schaffung (subjektiv für den Schüler) neuer mathematischer Objekte oder der Herausarbeitung neuer Ideen, siehe z. B. [213], [265], S. 221–228 und [268], S. 19–20.
- (3) *Visualisierungen* sowohl geometrischer als auch nichtgeometrischer Objekte und Zusammenhänge mithilfe des Computers erschließen anschauliche Zugänge und Betrachtungsweisen für Themengebiete, die mit klassischen Hilfsmitteln kaum veranschaulicht werden konnten und ermöglichen die Betrachtung bisher im Unterricht nicht auftretender Gegenstandsbereiche.<sup>109</sup> Im Zusammenhang mit dem experimentellen Arbeiten ermöglichen visuelle „was-wäre-wenn“-Analysen mithilfe von dynamischer Geometriesoftware (DGS) oder Echtzeit-Grafiksystemen sowie durch gezielte Parameteränderungen neue Wege der Erkenntnisgewinnung.
- (4) Der Computer ist ein wichtiges Hilfsmittel für die *Modellierung* realer Sachverhalte (vgl. Abschnitt 1.4.6). Teilweise können auch Modellierungen von Compu-

<sup>107</sup>Insbesondere für den Bereich der Elementargeometrie wurden in diesem Sinne sehr viele Vorschläge unterbreitet. Aber auch z. B. die Entdeckung von Gesetzmäßigkeiten in Zahlenfolgen (ein beliebtes Beispiel ist die Fibonacci-Folge) ist eine interessante Anwendung computerunterstützten experimentellen Arbeitens, siehe z. B. [265], S. 124–133.

<sup>108</sup>Der Begriff der Kreativität wird mitunter auf künstlerische Aspekte reduziert. In der vorliegenden Arbeit erfolgt seine Verwendung im Sinne einer Auffassung, die von WETH folgendermaßen charakterisiert wurde: „*Ein Produkt (eine Idee) soll im Mathematikunterricht als kreativ akzeptiert werden, wenn sie für den Schüler subjektiv neu ist oder neuartige Elemente enthält und wenn ein sinnvoller Beitrag zu einer Problemlösung gesehen wird oder selbst Problemstellungen erzeugt, welche einen reichhaltigen mathematischen Kontext eröffnen*“ ([265], S. 221).

<sup>109</sup>Die Verfügbarkeit des Computers zu Visualisierungszwecken hat eine recht breite Diskussion über didaktische Funktionen von Visualisierungen im Mathematikunterricht ausgelöst, siehe u. a. [43], [126], [127], [128], [130] und [202].

teranwendungen, die wiederum mithilfe des Computers vorgenommen werden, interessante Gegenstände des Mathematikunterrichts sein.<sup>110</sup>

- (5) Durch die Nutzung des Computers kann der Anteil der im Mathematikunterricht für *Berechnungen* aufgewendeten Zeit reduziert werden, was weitreichende Möglichkeiten in sich birgt<sup>111</sup> und didaktische Konsequenzen nach sich zieht (siehe z. B. [157], S. 204f.). So stellte SCHMIDT bereits 1993 für das Stoffgebiet Analytische Geometrie fest, dass durch die Nutzung von CAS „*die Ausführung algebraischer Kalküle zugunsten des Verstehens der Verfahren und der Anwendung und Interpretation derselben ... an Gewicht verliert*“ ([218], S. 27). HENN formulierte 2004, dass „*das starke Gewicht der Vermittlung und Anwendung von Kalkülen ... zu einer Sicht von ‚Mathematik als Produkt‘ führt. Der Paradigmenwechsel zu ‚Mathematik als Prozess‘ ... kann durch ... CAS gefördert werden*“ ([109], S. 218). Es ergeben sich daraus Konsequenzen für die Anforderungen in Klausuren und Abiturprüfungen, die aktuell vielfach noch zum überwiegenden Teil aus „Rechenaufgaben“ bestehen (vgl. 1.4.2). Einige Ansätze für in geeigneter Weise gestaltete Abituraufgaben sind u. a. in den EPA 2002 ([136]) zu finden.
- (6) Der Einsatz des Computers verschafft der Herausbildung eines Grundverständnisses für *algorithmisches Arbeiten* (das seit jeher Gegenstand des Mathematikunterrichts ist) einen höheren Stellenwert und – durch die Entwicklung von Algorithmen und ihre Umsetzung in kleine Programme bzw. Module durch Schüler – auch neue Möglichkeiten.<sup>112</sup> Zwei Aspekte algorithmischen Arbeitens seien hervorgehoben:
  - *Iterative und rekursive Denk- und Arbeitsweisen* bilden eine Grundlage jeglichen Programmierens, sind aber auch Bestandteile von Problemlöseprozessen im Mathematikunterricht und u. a. im Zusammenhang mit Zahlenfolgen und numerischen Näherungsverfahren von Bedeutung. Auch Rekursionen geometrischer Konstruktionen (wie bei der Konstruktion fraktaler Gebilde oder der Generierung bestimmter Kurven und Flächen) stellen interessante Möglichkeiten algorithmischen Arbeitens im Mathematikunterricht dar.
  - *Modulares Arbeiten* im Sinne des „Auslagerns von Können und Wissen“ ([265], S. 37) ist eine kognitive Strategie im Mathematikunterricht, die bereits ohne die Verwendung des Computers von Bedeutung ist – so werden gewisse Prozeduren wie Lösungskalküle oder geometrische Konstruktionen

<sup>110</sup>Beispiele finden sich u. a. in [116], S. 310–319 für die digitale Audiosignalverarbeitung und in [192] für die Computertomographie. Dabei werden Computeranwendungen zum Unterrichtsgegenstand, während der Computer bei den meisten Einsatzfällen im MU die Rolle eines Werkzeugs hat.

<sup>111</sup>So werden Modellierungen realer Sachverhalte, die aufwändige Berechnungen erfordern, durch den Einsatz von Tabellenkalkulations- oder Computeralgebrasystemen überhaupt erst möglich.

<sup>112</sup>Dazu ist nicht unbedingt die Beherrschung einer Programmiersprache im klassischen Sinne notwendig; von dieser Voraussetzung kann im Mathematikunterricht ohnehin nicht ausgegangen werden. Auch mit Tabellenkalkulationen und CAS lassen sich kleinere Algorithmen selbst entwickeln und umsetzen. So beschrieb GIEDING in [92] Möglichkeiten der Implementierung des Euklidischen Algorithmus in Excel, die auch durch Schüler der SI, die über keine Programmierkenntnisse verfügen, umsetzbar sind.

im Bedarfsfall aus dem Gedächtnis oder aus Aufzeichnungen bzw. Nachschlagewerken abgerufen.<sup>113</sup> Durch die Computernutzung können nun bestimmte „Prozeduren“ (z. B. lästige Routineaufgaben) auf ihn übertragen werden. Wünschenswert ist, dass Schülern modulare Arbeitsweisen verständlich werden, indem sie selbst Prozeduren erstellen, diese als Module speichern und für andere Aufgaben wiederverwenden. Dies ist natürlich in Programmiersprachen, aber auch in CAS möglich.<sup>114</sup> Eine besonders intuitive Möglichkeit für Schüler, bereits in der Mittelstufe selbst Module zu erstellen, bieten DGS.<sup>115</sup>

- (7) Einen höheren Stellenwert erhalten durch die Einbeziehung des Computers und die (zumindest ansatzweise) Berücksichtigung von Arbeitsweisen der Informatik *Datentypen und Datenstrukturen* im Mathematikunterricht (siehe u. a. [112], S. 15 und [262], S. 135). Im Hinblick auf die Thematik der vorliegenden Arbeit ist dieser Aspekt insbesondere hinsichtlich von Sichtweisen auf Vektoren und Matrizen interessant.<sup>116</sup> Vor allem auf Konsequenzen für die Behandlung des Vektorbegriffs, die sich aus der in der Informatik gebräuchlichen Auffassung von Vektoren ergeben, ist in dieser Arbeit noch einzugehen (siehe Abschnitt 3.3.1).

Scharfe Trennlinien zwischen den genannten Aspekten können nicht gezogen werden; vielmehr überlappen sich bei fast jeder Nutzung des Computers im Mathematikunterricht mehrere der Aspekte (1)–(5). So stehen Modellierungen oft im Zusammenhang mit Berechnungen, Experimenten und Visualisierungen.

Bei den Aspekten (1), (2), (3), (5) und weitgehend bei (4) steht hauptsächlich die Nutzung des Computers als *Werkzeug* (durchaus im Sinne einer „black box“) im Mittelpunkt. Hingegen wird bei (4), soweit Computeranwendungen modelliert werden, sowie vor allem bei (6) und – oft damit im Zusammenhang – (7) die Arbeitsweise des Computers (beziehungsweise einzelner seiner Elemente und Anwendungen) zum *Unterrichtsgegenstand* des Mathematikunterrichts. Eine verstärkte Berücksichtigung dieser informatikbezogenen Aspekte (die im Mathematikunterricht im Gegensatz zur Nutzung als Werkzeug eine vergleichsweise geringe Rolle spielen) ist m. E. wünschenswert, gehören doch der Computer und seine „Produkte“ mittlerweile zur „Lebenswirklichkeit“ (siehe Abschnitt 1.4.6) und die Grundlagen ihrer Funktionsweise liegen zu weiten

<sup>113</sup>WEIGAND begründete in [263], dass modulares Arbeiten eine typische mathematische Arbeitsweise darstellt und bereits in den „Elementen“ des EUKLID auftritt. In [262] führte er aus, dass „*ein Metawissen über prozedurales Arbeiten ... im Mathematikunterricht insbesondere für den Aufbau heuristischer Strukturen hilfreich ist.*“

<sup>114</sup>Eine Reihe von Vorschlägen für modulares Arbeiten unter Nutzung von CAS im Zusammenhang mit Standardthemen des Mathematikunterrichts unterbreitete E. LEHMANN, der für die Module die Bezeichnung „CAS-Bausteine“ verwendete, siehe u. a. [150] und Abschnitt 2.1.5 in [153].

<sup>115</sup>Geometrische Konstruktionen (die sich als „Prozeduren“ auffassen lassen) können in den meisten DGS als Makros bzw. „Custom Tools“ gespeichert werden. Einmal durch den Schüler durchgeführte Konstruktionen bereichern so den „Werkzeugvorrat“ der entsprechenden Software und stehen fortan zur Verfügung. (Zu modularem Konstruieren siehe auch z. B. [268].)

<sup>116</sup>In den neuen Hamburger Rahmenplänen wurde versucht, der Verwendung von Vektoren und Matrizen in der Informatik Rechnung zu tragen, siehe S. 31.



Teilen in der Mathematik.<sup>117</sup> Für die Thematik der vorliegenden Arbeit sind, wie bereits erwähnt wurde, sowohl Aspekte der Computernutzung als auch die Thematisierung mathematischer Grundlagen einer wichtigen Computeranwendung von Bedeutung.

Neben den genannten Aspekten hat der Computereinsatz im Mathematikunterricht weitreichende *Auswirkungen auf die Unterrichtsgestaltung*, die Veränderungen in bereits in Abschnitt 1.4.8 als wünschenswert herausgearbeitete Richtungen bedingen. So heißt es in der bereits mehrfach erwähnten Expertise [29]:

*„Schülerbezogene Arbeitsformen wie Partner-, Gruppen- und Projektunterricht, Förderung von Selbstständigkeit und Selbstverantwortung, entdeckender Unterricht, umwelterschließender Unterricht, alles das sind Forderungen, die zumindest seit der Reformpädagogik an die Schule herangetragen werden. Erfahrungen zum Computereinsatz geben heute zu der Hoffnung Anlass, dass neue Technologien ein Katalysator für eine solche ‚neue Unterrichtskultur‘ sein können.“*

WEIGAND und WETH formulierten, dass *„der computerunterstützte Unterricht fast zwangsläufig vom traditionellen, lehrerzentrierten Unterricht weg(führt), was nicht nur positiv, sondern auch kritisch zu reflektieren ist“* ([265], S. XIII).<sup>118</sup> Möglichkeiten selbstgesteuerter mathematischer Aktivitäten der Schüler durch die Arbeit am Computer wurden vielfach hervorgehoben, stellvertretend sei hier auf HENN ([109], S. 217) verwiesen. Ebenfalls eine für das Arbeiten mit dem Computer prädestinierte Unterrichtsform sind Unterrichtsprojekte, worauf u. a. BAUMANN in [10] näher einging.

Sowohl zu eigenständigem und (unter Beachtung vorgegebener Aufgaben und Ziele) selbstbestimmtem Arbeiten als auch zu Unterrichtsprojekten am Computer werden in der vorliegenden Arbeit im Zusammenhang mit dem Einsatz von Computergrafik-Software Vorschläge unterbreitet und Erfahrungen geschildert.

---

<sup>117</sup>Ich möchte damit nicht für die Einführung „klassischer“ Programmiersprachen im Mathematikunterricht plädieren, die einen hohen Einarbeitungsaufwand erfordern. Diese in den achtziger Jahren mitunter erhobene Forderung ist aus Zeitgründen und aufgrund der mittlerweile erfolgten Etablierung eines eigenständigen Unterrichtsfaches Informatik nicht mehr aktuell. Jedoch ermöglichen es genuin mathematische Inhalte, Grundlagen der Funktionsweise von Computeranwendungen zu thematisieren. Auch die Nutzung und Reflexion elementarer „Programmiermöglichkeiten“ derjenigen Software, die im Unterricht eingesetzt wird (hauptsächlich CAS, DGS und Tabellenkalkulation), erscheint wünschenswert; auf entsprechende Ansätze unter Verwendung von Grafiksoftware wird in dieser Arbeit noch eingegangen. Schließlich könnten es übergreifende Projekte mit dem Fach Informatik ermöglichen, mathematische Grundlagen von Computeranwendungen im Mathematikunterricht zu thematisieren und auf dieser Grundlage im Informatikunterricht Algorithmen unter informatischen Gesichtspunkten zu „verfeinern“ und zu implementieren. Dies wird allerdings dadurch erschwert, dass meist nicht alle Schüler einer Klasse das Fach Informatik belegen.

<sup>118</sup>Wohl jeder Lehrer, der bereits computerunterstützten Unterricht erteilt hat, wird den Autoren zustimmen. Enormen Chancen kreativen und selbstständigen Arbeitens der Schüler steht die Notwendigkeit einer besonders sorgfältigen und durchdachten Unterrichtsplanung und -vorbereitung gegenüber. Stundenverläufe, die den angestrebten Zielen nicht entsprechen, sind verhältnismäßig schwer korrigierbar, wenn die Schüler an Computern arbeiten. U. a. mit dem Ziel, die Arbeit der Schüler am Computer trotz selbstständiger, individueller Lernmöglichkeiten zielführend zu gestalten, wurde das Konzept der „Lernumgebungen“ entwickelt, siehe z. B. [157], S. 246–262.

### 1.5.2 Computernutzung im Stoffgebiet Analytische Geometrie

Hinsichtlich der Nutzung des Computers im Unterricht der linearen Algebra / analytischen Geometrie wurden im Verlauf der vergangenen Jahrzehnte viele Vorschläge unterbreitet, die sich folgendermaßen (grob) kategorisieren lassen:

- Die Nutzung von CAS für das *Lösen von Gleichungssystemen* und damit u. a. für die *Berechnung von Schnittpunkten und -geraden* kann die für Rechnungen aufgewendete Zeit reduzieren und ermöglicht die Behandlung rechnerisch komplexerer Beispiele, siehe (5) auf S. 57. Inhaltliche Veränderungen ergeben sich dadurch nicht; allerdings wurden Vorschläge zur CAS-Nutzung für Berechnungen meist mit anderen Aspekten des Computereinsatzes verbunden. Dazu gehören komplexere Anwendungen der linearen Algebra sowie die Modellierung realer Situationen.
- Die Nutzung von CAS oder spezieller Software<sup>119</sup> für die *Visualisierung* geometrischer Objekte und Lagebeziehungen ist Gegenstand vieler Vorschläge.<sup>120</sup> Teilweise zielen diese darauf ab, die im Unterricht standardmäßig behandelten Geraden und Ebenen (teilweise auch Kreise und Kugeln) sowie ihre Lage zu veranschaulichen (siehe z. B. [3], [230] und [259]). Auch die Visualisierung von Inhalten und Verfahren der linearen Algebra wie Eigenwerten, Eigenvektoren, linearen Gleichungssystemen und des GAUSS-Algorithmus wurde unter didaktischen Gesichtspunkten diskutiert (siehe z. B. [162] und [214]).
- Wie bereits in Abschnitt 1.4.5 erwähnt wurde, ist die Computernutzung (insbesondere zu Visualisierungszwecken) Bestandteil vieler Ansätze zur *Behandlung interessanter Kurven und Flächen* im Unterricht.<sup>121</sup> Die experimentellen und visuellen Möglichkeiten, die der Computer eröffnet, sind eine wichtige Voraussetzung vieler *Objektstudien* (siehe hierzu auch [247], S. 153f. sowie den Vorschlag [151] von LEHMANN zu computerunterstützten „Impressionen an Kreisen“).
- Einige Autoren entwickelten Konzepte zur Behandlung von *Matrizen* unter Verwendung des Computers (siehe u. a. [152], [153] und [280]) und stellten Bezüge zu nichtgeometrischen Anwendungen, zur Informatik (Matrizen als Datenstrukturen) und zur Beschreibung geometrischer Abbildungen her. In diesem Zusammenhang wurde auch die Behandlung von Elementen der Computergrafik ins Auge gefasst.

*Beziehungen zwischen der analytischen Geometrie und der Computergrafik* wurden seit längerem in didaktischen Überlegungen aufgegriffen. Auf diesbezügliche Ansätze und Konzepte wird nach einer in Kapitel 2 erfolgenden Darstellung der Funktionsweise und der mathematischen Grundlagen der 3D-Computergrafik in Abschnitt 3.1 eingegangen.

<sup>119</sup>Für Visualisierungen in der analytischen Geometrie wurden spezielle Programme entwickelt, die effektivere Veranschaulichungen ermöglichen sollten als einfache CAS. Durch die Entwicklung der Grafikfunktionalitäten von CAS wie Maple und MuPAD haben diese speziellen Programme jedoch etwas an Bedeutung verloren; siehe dazu die Abschnitte 2.7.5, 2.7.6 und 3.4.2.

<sup>120</sup>Zur Spezifik von Veranschaulichungen in der analytischen Geometrie siehe Abschnitt 1.4.4.

<sup>121</sup>Dabei kommen vor allem die auf S. 56ff. dargestellten Aspekte (1), (2) und (3), teilweise auch (4) und (5) sowie im Falle der Bézierkurven (6) zum Tragen.

## 1.6 Zusammenfassung, Zwischenfazit und Ausblick

### 1.6.1 Mögliche Entwicklungsrichtungen des Unterrichts in analytischer Geometrie

In diesem Kapitel, vor allem in Abschnitt 1.4, wurden Vorschläge skizziert, Probleme des Unterrichts in analytischer Geometrie bzw. linearer Algebra zu lösen und den allgemeinbildenden Wert sowie die Attraktivität des entsprechenden Stoffgebietes zu erhöhen. Natürlich werden nicht alle dieser Vorschläge in einem Kurs gleichermaßen zum Tragen kommen können, ihre Berücksichtigung hängt von zu wählenden Schwerpunkten ab. Prinzipiell sind m. E. drei sinnvolle Entwicklungsrichtungen des Unterrichts in linearer Algebra / analytischer Geometrie denkbar:

- (1) Durchführung eines Kurses mit dem Schwerpunkt „Lineare Algebra“, bei dem (hauptsächlich nichtgeometrische) Anwendungen linearer Gleichungssysteme sowie des Vektor- und evtl. Matrixkalküls im Vordergrund stehen (siehe 1.2.4).<sup>122</sup>
- (2) Untersuchung interessanter geometrischer Objekte, vor allem Kurven, mit Mitteln der analytischen Geometrie im Zweidimensionalen unter Verwendung klassischer Koordinaten-, teilweise auch vektorieller Methoden (vgl. 1.4.5). Bei einem derartigen Vorgehen bieten sich gute Möglichkeiten, Verbindungen zur Analysis herzustellen (siehe 1.4.7).<sup>123</sup>
- (3) Schwerpunktmäßige Behandlung einer analytischen Raumgeometrie bei gleichzeitiger geometrischer „Anreicherung“. Gegenüber der heute dominierenden vektoriellen Raumgeometrie, bei der fast ausschließlich lineare Objekte mit den Mitteln der Vektorrechnung untersucht werden (vgl. 1.3), sollten interessantere geometrische Objekte des Raumes (wie Flächen und Raumkurven) in den Unterricht einbezogen werden; dazu ist eine flexiblere Wahl der Beschreibungsmethoden erforderlich. Neben der Anwendung vektorieller ist auch die Nutzung koordinatengeometrischer Vorgehensweisen sinnvoll,<sup>124</sup> wobei ein Anknüpfen an die Elementargeometrie und die Herstellung von Verbindungen zur Analysis (mindestens zu elementaren Funktionen) wünschenswert erscheint. Geometrisch orientierte Anwendungen und Modellierungen können den Unterricht wesentlich bereichern.

Grundsätzlich besteht kein Widerspruch zwischen diesen drei Entwicklungsrichtungen. Allerdings wird das Maß an zur Verfügung stehender Zeit die Entscheidung zwischen einer geometrischen und einer auf die Behandlung anspruchsvoller nichtgeometrischer Anwendungen ausgerichteten Gestaltung des Stoffgebietes erfordern, wenngleich Einzelaspekte des jeweils anderen Ansatzes aufgegriffen werden können. Auch zwischen einer

<sup>122</sup>Einige Rahmenpläne bieten eine derartige Schwerpunktsetzung als Wahlmöglichkeit an (vgl. 1.3).

<sup>123</sup>Wie bereits in Abschnitt 1.3 kurz ausgeführt wurde, enthalten die neuen Curricula der Jahrgangsstufe 11 für Berlin und Hamburg Ansätze, die in diese Richtung gehen.

<sup>124</sup>In Grundkursen ist bei einem engen zeitlichen Rahmen für die analytische Geometrie eine Beschränkung auf Koordinatengeometrie unter Verzicht auf die Behandlung von Vektoren in Erwägung zu ziehen. Diese Möglichkeit wird zunehmend ins Auge gefasst, u. a. um die unter Zeitdruck erfolgende fast vollständige Beschränkung auf die Behandlung des Vektorkalküls zu vermeiden (vgl. [247], S. 157).

Behandlung ebener und räumlicher analytischer Geometrie besteht selbstverständlich kein Widerspruch. Mehrere Beispiele in dieser Arbeit werden zeigen, dass die Reduzierung räumlicher auf ebene Probleme und die Rückübertragung gefundener Lösungen auf dreidimensionale Objekte eine wichtige Strategie zur Lösung raumgeometrischer Aufgaben und Probleme ist. Die Kombination einer ebenen analytischen Geometrie in Klasse 11 und eines Stoffgebietes zur analytischen Raumgeometrie in der Kursstufe (die nach den Rahmenplänen einiger Bundesländer nahe liegt) ist m. E. eine sinnvolle Kombination zur Behandlung der analytischen Geometrie in der Sekundarstufe II.

Die in dieser Arbeit unterbreiteten Vorschläge und Konzepte zielen auf den Schwerpunkt (3) ab, sehen also die analytische Behandlung von Elementen der Raumgeometrie vor, wobei interessante geometrische Objekte einbezogen und Akzente auf Anwendungen im Bereich der 3D-Computergrafik gesetzt werden. M. E. ist eine derartige Schwerpunktsetzung unter anderem aus dem Grunde sinnvoll, dass die Geometrie des Raumes bereits im Mathematikunterricht der S I unterrepräsentiert ist.

### **1.6.2    Computergrafik im Unterricht der analytischen Geometrie – Überblick über Potenzen und Ziele**

Die Bedeutung computergrafischer Anwendungen (wozu im weiteren Sinne auch CAD zählt) ist in den vergangenen Jahren enorm gewachsen<sup>125</sup> und Computergrafik wurde zu einem Teil der „Lebenswirklichkeit“ von Jugendlichen. Es erscheint daher geboten, ausgehend vom gegenwärtigen Stand ihrer Entwicklung eine Analyse der Inhalte der Computergrafik hinsichtlich ihrer Einbeziehung in das Stoffgebiet Analytische Geometrie vorzunehmen und geeignete Unterrichtskonzepte hierfür zu entwickeln. So forderte SCHUPP 2000 in seinem bereits mehrfach erwähnten Beitrag [234]:

*„Wir stehen am Beginn des piktoralen Zeitalters. Die Schule ... sollte versuchen, die enormen visuellen Potenzen der neuen Medien zu enkultivieren. Geometrieunterricht kann und muss dazu einen wichtigen Beitrag leisten, nämlich das verweilende Arbeiten am Computerbild inszenieren, seine Konstruktion, Fokussierung, Anreicherung, Exploration, Dynamisierung, Variation ermöglichen und einfordern. Mit dem bloßen Visualisieren anders und anderwärts geklärter Sachverhalte ... ist noch nicht viel erreicht, zumal wir heute wissen, dass Bilder nicht so sehr als methodische Hilfe denn als eine wesentliche Komponente des Wissens selbst aufgefasst werden müssen.“* ([234], S. 54)

Für die Weiterentwicklung des Unterrichts in analytischer Geometrie und die Überwindung wesentlicher Defizite des Unterrichts in diesem Stoffgebiet bietet die Einbeziehung von Elementen der 3D-Computergrafik eine Reihe wertvoller Möglichkeiten:<sup>126</sup>

---

<sup>125</sup>Festzustellen, dass die Bedeutung der analytischen Geometrie aufgrund dieser Anwendungen zugenommen hat, ist sicherlich nicht übertrieben.

<sup>126</sup>Diese Möglichkeiten werden hier vorab genannt, ohne dass fundierte Begründungen erfolgen – dies ist erst nach der in Kapitel 2 erfolgenden Analyse der mathematischen Grundlagen der 3D-Computergrafik möglich. Die hier aufgeführten Potenzen der Computergrafik für den Unterricht geben aber bereits eine grobe Orientierung für die folgenden Kapitel.

- Durch die Ergänzung algebraischer Untersuchungen geometrischer Objekte und Relationen durch computergrafische Visualisierungen lässt sich eine „Geometrisierung“ und Veranschaulichung des Unterrichts im Stoffgebiet Analytische Geometrie erreichen.
- Die Verwendung computergrafischer Software ermöglicht heuristische und experimentelle Wege der Untersuchung geometrischer Objekte und damit die verstärkte Einbeziehung dieser Arbeitsweisen in den Unterricht.
- Durch den mithilfe von 3D-Grafiksoftware möglichen visuellen und experimentellen Zugang eröffnen sich Möglichkeiten, interessante nichtlineare geometrische Objekte zu betrachten und damit eine größere Formenvielfalt zu erreichen.
- Die Schüler erfahren die enorme Anwendungsrelevanz der analytischen Geometrie auf einem Gebiet, das für sie attraktiv ist. Sie können den Nutzen ihres mathematischen Wissens bei der Gestaltung von Computerbildern erleben, indem sie vom „Spielen“ zu einem systematischen Vorgehen finden.
- Es bestehen sehr gute Möglichkeiten für das aktive und selbstständige Arbeiten der Schüler. Der bei vielen Schülern schnell entstehende Wunsch nach der Gestaltung interessanter Computergrafiken und -animationen kann die Beschäftigung mit mathematischen Aspekten, die dazu notwendig sind, motivieren.
- Das Herstellen stoffgebiets- und fächerübergreifender Bezüge (u. a. zur Analysis, Physik, Informatik und Kunst) wird erleichtert oder erst ermöglicht.
- Nicht zuletzt kann der ästhetische Reiz dreidimensionaler computergrafischer Darstellungen eine Motivierung der Schüler für den Unterricht bewirken.

Die 3D-Computergrafik kann nicht nur als *Hilfsmittel* für den Unterricht dienen, sondern auch als *Unterrichtsgegenstand* den Schülern die Bedeutung und Nützlichkeit grundlegender Inhalte und Arbeitsweisen der analytischen Geometrie an einem praxisrelevanten, interessanten Gegenstand verdeutlichen. Somit ergeben sich folgende Hauptaspekte der Einbeziehung der 3D-Computergrafik in den Mathematikunterricht:

- Computer (3D-Grafiksoftware) als *Werkzeug* zur Visualisierung
  - der traditionellen Inhalte des Unterrichts in analytischer Geometrie (Vektoren, Geraden und Ebenen) sowie einfacher geometrischer Körper,
  - „neuer“ Unterrichtsinhalte – Erschließung zur Zeit nicht behandelter geometrischer Objekte wie Kurven und Flächen.
- Thematisierung (zumindest ausgewählter Aspekte) der Funktionsweise der 3D-Computergrafik und -animation als *Unterrichtsgegenstand*. Hierbei kommen Vorgehensweisen der Modellierung in mehrfacher Hinsicht (und in unterschiedlicher Bedeutung des Wortes) zum Tragen, vgl. Abschnitt 1.4.6.

Die Erschließung diesbezüglicher Potenziale, die eine fundierte Analyse der mathematischen Grundlagen der 3D-Computergrafik und der Bezüge dieser Grundlagen zu anwendungspraktischen Erfordernissen voraussetzt, ist das Ziel der vorliegenden Arbeit.



## Kapitel 2

# Mathematische Grundlagen der 3D-Computergrafik und ihre Umsetzung in Grafiksoftware

In diesem Kapitel werden die wichtigsten Herangehens- und Arbeitsweisen der dreidimensionalen Computergrafik sowie ihre mathematischen Grundlagen überblicksartig dargestellt. Damit soll eine Basis für in den Kapiteln 3 und 4 anzustellende Überlegungen geschaffen werden, welche Elemente der 3D-Computergrafik sich für die Einbeziehung in den Unterricht der analytischen Geometrie eignen. Obwohl dafür nicht alle Aspekte der Computergrafik (CG) gleichermaßen in Frage kommen, wird hier ein Überblick über alle für die Erzeugung computergrafischer Darstellungen notwendigen Arbeitsschritte gegeben, da erst auf der Grundlage eines derartigen Gesamtüberblicks eine Bewertung aus didaktischer Sicht erfolgen kann.<sup>1</sup> Allerdings folgt auch bereits das vorliegende Kapitel einer Schwerpunktsetzung zugunsten mathematischer Inhalte, deren Behandlung in der Sekundarstufe II prinzipiell möglich erscheint. Themenbereiche, für die das offensichtlich nicht zutrifft, werden nur sehr knapp dargestellt.<sup>2</sup>

Die Akzentsetzung und die inhaltliche Strukturierung dieses Kapitels orientieren sich hauptsächlich an Vorgehensweisen bei der Erstellung von Grafiken unter *Anwendung* computergrafischer Software. Es soll deutlich werden, dass das Verständnis mathematischer, vor allem geometrischer Grundlagen wesentlich für die bewusste und erfolgreiche Nutzung von CG-Software ist. Hinsichtlich der Zielsetzungen der vorliegenden Arbeit ist es notwendig, Verknüpfungen zwischen mathematischen und anwendungspraktischen Aspekten der Computergrafik herzustellen.<sup>3</sup>

---

<sup>1</sup>In der mathematikdidaktischen Literatur wurden die Grundlagen der Computergrafik (mit Ausnahme der bereits älteren Arbeit [90] von GIEDING) bisher nur sehr punktuell betrachtet.

<sup>2</sup>Es sei dazu auf die umfangreiche einschlägige Literatur, insbesondere auf [11], [31], [34], [57], [64], [78], [79], [94], [102], [124], [125], [164], [169], [185], [189], [248], [256], [257], [258] und [282] verwiesen.

<sup>3</sup>In der vorhandenen Literatur fehlen derartige Verknüpfungen bislang weitgehend bzw. werden nur indirekt sichtbar. Ein Teil der Fachliteratur zur Computergrafik behandelt mathematische Grundlagen vorrangig mit dem Ziel der Entwicklung von Algorithmen und der Programmierung von Software, während ein anderer Teil Anwendungen vor allem künstlerisch orientiert oder hinsichtlich der Bedienung von Software beschreibt, dabei jedoch mathematische Aspekte kaum berücksichtigt.

Aspekte der *Implementierung von Algorithmen* der Computergrafik und ihrer *programmiertechnischen Umsetzung* sind nur hinsichtlich der mathematischen Grundlagen Gegenstand dieses Kapitels, wenngleich an einigen Stellen Algorithmen beschrieben und anhand einfacher Beispiele in Skripte bzw. kleine Programme umgesetzt werden.<sup>4</sup> Dabei erfolgt eine Beschränkung auf geometrische Konstruktionen und Verfahren sowie Algorithmen zur Bildberechnung (Rendering-Verfahren) unter weitgehendem Verzicht auf die Darstellung von in der Computergrafik verwendeten Datenstrukturen und Daten-sortieralgorithmen. Diese hätten den Rahmen der vorliegenden Arbeit sowohl inhaltlich als auch hinsichtlich des Umfangs überschritten.

Um Zusammenhänge zwischen den mathematischen Grundlagen der 3D-Computergrafik und Belangen der praktischen Anwendung in geeigneter Grafiksoftware zu verdeutlichen, enthalten nahezu alle Abschnitte des vorliegenden Kapitels illustrierende Beispiele für die Generierung bzw. Nutzung der beschriebenen mathematischen Objekte und Verfahren. Die meisten dieser Beispiele wurden unter Verwendung der skriptgesteuerten 3D-Grafiksoftware POV-Ray erstellt.<sup>5</sup> Da die Syntax von POV-Ray mit der von Computeralgebrasystemen vergleichbar und recht intuitiv ist, sind ausführliche Erklärungen der Beispiele oftmals nicht notwendig. Die dargestellten Skriptzeilen können auch als effizient geschriebene „Handlungsanweisungen“ für die Bedienung (mausgesteuerter) Grafiksoftware gelesen werden. So bedeutet z. B.

```
translate <x,y,z>,
```

dass ein Objekt um die Längen  $x$ ,  $y$  und  $z$  entlang der entsprechenden Achsen verschoben (bzw. mit der Maus „gezogen“) wird. Die Darstellung von Skriptanweisungen kann auch die Eingabe von Werten in Menüs und Dialogfelder effektiv beschreiben.

Am Ende dieses Kapitels werden verschiedene Kategorien von Programmen beschrieben, mit denen sich dreidimensionale computergrafische Darstellungen erzeugen lassen. Dabei handelt es sich zum großen Teil um eine Zusammenfassung und Ergänzung der innerhalb des gesamten Kapitels anhand von Beispielen zu den Arbeitsschritten der 3D-Computergrafik aufgezeigten Funktionen und Verfahren verschiedener Softwarepakete.

Da 3D-Computergrafiken letztendlich in fast allen Fällen in zweidimensionale Bilder umgewandelt (gerendert) und als solche betrachtet werden, beginnt das Kapitel mit einer kurzen Darstellung der Möglichkeiten, Bilder und Grafiken im Computer zu repräsentieren und zu bearbeiten sowie Farben mathematisch zu beschreiben.

<sup>4</sup>Die Skripte (hauptsächlich POV-Ray-Szenenbeschreibungen, kleinere Skripte in der Programmiersprache Python sowie innerhalb des CAS MuPAD erstellte Prozeduren) sind nur in für das jeweilige Thema relevanten Teilen abgedruckt. Die vollständigen Skripte lassen sich von der Internetseite <http://www.afiller.de/habil> herunterladen. Dort steht eine Datei **Filler-Habil-Dateien.zip** zur Verfügung, die alle in der vorliegenden Arbeit mit ihren Dateinamen erwähnten Skripte, Dateien und zusätzliche Materialien enthält (siehe dazu Anhang D, S. 365). Nach dem Entpacken dieser .zip-Datei wird eine Reihe von Ordnern angelegt. Die Namen der Ordner ergeben sich aus der Nummerierung der Abschnitte der Arbeit – so liegen z. B. Dateien zu Abschnitt 2.1 in dem Ordner 2-1.

<sup>5</sup>POV-Ray ist ein hochwertiges 3D-Grafikprogramm, das sich gut für die Verdeutlichung mathematischer Grundlagen der 3D-Computergrafik eignet, da Grafiken durch eine Szenenbeschreibungssprache erzeugt werden. POV-Ray ist unter <http://www.povray.org> frei verfügbar.



## 2.1 Grundlegende Verfahren der Computergrafik und der Bildbearbeitung

Bilder und Grafiken werden prinzipiell durch zwei Aspekte gekennzeichnet: Formen und Farben. Daher sind in der Computergrafik einerseits Formen, Konturen, Verläufe – also die topologischen Informationen in Bildern – und andererseits Farben mathematisch zu beschreiben und für die Bildschirmdarstellung bzw. den Druck aufzubereiten. In diesem Abschnitt werden zunächst die Repräsentation von Bildstrukturen und Grafikobjekten und dann die mathematische Beschreibung von Farben überblicksartig dargestellt.

### 2.1.1 Repräsentation und Verarbeitung von Bildern und Grafiken im Computer – Raster- und Vektorgrafik

Für die Darstellung und Bearbeitung von Bildern und Grafiken mithilfe von Computern gibt es zwei grundsätzlich verschiedene Vorgehensweisen: die *Raster-* bzw. *Pixelgrafik*, die Bilder durch Aufteilung in gleich große Quadrate bzw. durch ein Punktraster erfasst und darstellt, sowie die *Vektorgrafik*, welche Grafikbestandteile als geometrische Objekte beschreibt. Für die 3D-Computergrafik sind beide Herangehensweisen von Bedeutung.

#### Rastergrafik – Beschreibung von Bildern durch Pixel

In der digitalen Fotografie, Bilderfassung und -bearbeitung sowie bei der Bildausgabe auf Monitoren und Druckern werden Bilder durch Helligkeits- bzw. Farbwerte einzelner Bildpunkte (Pixel, Abkürzung für picture elements) repräsentiert.<sup>6</sup> Ein Bild wird dazu in eine möglichst große Anzahl von Quadraten unterteilt und es werden die Helligkeitswerte der Mittelpunkte dieser Quadrate erfasst. Unter Pixeln werden also die „Elementarquadrate“ eines Bildes oder die entsprechenden Mittelpunkte verstanden.<sup>7</sup>

Anhand der Unterteilung in Pixel kann jedes Graustufenbild durch eine Matrix beschrieben werden, welche die Helligkeitswerte aller Pixel des Bildes enthält. Die folgende Matrix  $M$  repräsentiert das in Abb. 2.1 dargestellte Bild.<sup>8</sup>

---

<sup>6</sup>Im Folgenden erfolgt zunächst eine Beschränkung auf Graustufenbilder; bei diesen wird jeder Punkt durch einen Helligkeitswert beschrieben. Bei farbigen Darstellungen müssen für jeden Punkt drei Helligkeitswerte erfasst werden, auf Grundlage derer Farben beschrieben werden können. Die dem zugrunde liegenden Farbmodelle werden in Abschnitt 2.1.2 beschrieben.

<sup>7</sup>Da realistische Bilder aus sehr vielen Pixeln bestehen, die mit bloßem Auge nicht erkennbar sind, ist die Frage, ob Pixel Quadrate, Punkte oder kleine Kreise sind, nebensächlich. Bei Bildschirmen besteht jedes Pixel aus drei Subpixeln – kleinen Kreisen (bzw. bei einigen Bauarten von Bildröhren Streifen), die in den Farben Rot, Grün und Blau leuchten. Aus diesen drei Grundfarben werden alle Farben erzeugt (siehe Abschnitt 2.1.2).

<sup>8</sup>Die Darstellung der Matrix eines „realistischen“ Fotos war aus Platzgründen nicht möglich. Für erkennbare Bilder werden mindestens einige Hundert Zeilen und ebenfalls einige Hundert Spalten benötigt; Bilder für hochwertige Druckerzeugnisse haben sogar eine Auflösung von 300–1200 Pixeln je Zoll (dots per inch, dpi), also ca. 118–472 Pixeln je cm. Die beschreibende Matrix eines 15 cm × 10 cm großen Bildes mit einer Auflösung von 300 dpi besteht bereits aus 1180 Zeilen und 1770 Spalten.

$$M = \begin{pmatrix} 0,37 & 0,94 & 0,89 & 0,60 & 0,18 & 0,07 & 0,10 & 0,07 \\ 0,18 & 0,60 & 0,89 & 0,91 & 0,59 & 0,23 & 0,22 & 0,11 \\ 0,05 & 0,26 & 0,70 & 0,92 & 0,92 & 0,64 & 0,33 & 0,12 \\ 0,03 & 0,27 & 0,80 & 0,80 & 0,92 & 0,94 & 0,76 & 0,38 \\ 0,01 & 0,12 & 0,30 & 0,32 & 0,53 & 0,83 & 0,98 & 0,78 \\ 0,00 & 0,01 & 0,04 & 0,05 & 0,12 & 0,36 & 0,66 & 0,63 \end{pmatrix}$$



Abbildung 2.1:  
Rastergrafik

Die Mathematica-Datei **Rastergrafik1.nb** (in dem Ordner 2-1; vgl. Anhang D, S. 365) liest ein Bild ein und die zeigt die zugehörige Matrix an.

Bildbearbeitung und -manipulation beruhen auf der Veränderung der den Bildern zugeordneten Matrizen (mit meist sehr vielen Zeilen und Spalten).<sup>9</sup> Einfache Bildbearbeitungsschritte wie das gleichmäßige Aufhellen oder Abdunkeln von Bildern sowie die Erzeugung von Negativbildern sind mit recht elementaren Operationen (z. B. Multiplikation von Matrizen mit reellen Zahlen) realisierbar. Abbildung 2.2 b) entstand in dem Computeralgebrasystem (CAS) Mathematica, indem die beschreibende Matrix  $M$  des in Abb. 2.2 a) dargestellten Bildes (mit  $236 \times 116$  Pixeln) von der Matrix mit ebenso vielen Zeilen und Spalten, deren sämtliche Elemente 1 sind, subtrahiert wurde:

$$M_{neg} = \begin{pmatrix} 1 & \dots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \dots & 1 \end{pmatrix} - M.$$

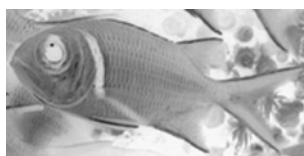
Weiter gehende Schritte der Bildbearbeitung (wie z. B. Weichzeichnen, Schärfen, Verminderung von Bildrauschen sowie diverse Spezialeffekte) erfordern Algorithmen, die über die bekannten Matrixoperationen hinausgehen. Dabei sind vor allem zwei Herangehensweisen von Bedeutung:

- Bei *Nachbarschafts-* bzw. *lokalen Operationen* werden für jedes Pixel die Werte der Pixel in einer näheren Umgebung betrachtet und bei der Berechnung der Helligkeitswerte berücksichtigt. Durch *Faltungsooperatoren* bzw. *Faltungsmatrizen* werden Pixeln Helligkeitswerte zugeordnet, die sich aus ihren eigenen ursprünglichen Helligkeiten sowie den Helligkeiten der Pixel in einer (mehr oder weniger weit ausgedehnten) Umgebung ergeben. So ordnet die Faltungsmatrix

<sup>9</sup>Die der Bildbearbeitung zugrunde liegenden mathematischen Operationen bleiben vor den Benutzern von Bildbearbeitungssoftware verborgen. Diese müssen lediglich aus dem sehr großen Funktionsumfang, den Programme wie Adobe Photoshop und Corel Photopaint bieten, die benötigten Befehle auswählen und geeignete Parameter (wie z. B. für die Helligkeit und den Kontrast) einstellen. In CAS wie Mathematica und Maple, die den Import von Bilddaten erlauben, ist es möglich, die Wirkung von Matrixoperationen auf Bilder sichtbar zu machen. Dabei fällt auf, dass CAS für einfache Bildbearbeitungsschritte ein Vielfaches der Zeit benötigen, in der die hoch optimierten Routinen von Bildbearbeitungsprogrammen erheblich komplexere Bearbeitungen vornehmen.



a)



b)

Abbildung 2.2: a) Ausgangsbild;  
b) zugehöriges Negativ

Datei: 2-1\RastergrafikFisch.nb

$$F_1 = \frac{1}{16} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} \quad \begin{array}{c|c|c} 0,37 & 0,94 & 0,89 \\ \hline 0,18 & \mathbf{0,60} & 0,89 \\ \hline 0,05 & 0,26 & 0,70 \end{array} \longrightarrow \begin{array}{c|c|c} \dots & \dots & \dots \\ \hline \dots & \mathbf{0,56} & \dots \\ \hline \dots & \dots & \dots \end{array}$$

jedem Pixel den mit den Matrixelementen von  $F$  gewichteten Mittelwert der Helligkeitswerte der Pixel eines aus 9 Pixeln bestehenden Quadrates zu. Für das oben rechts angegebene Beispiel ergibt sich bei der Anwendung der Faltungsmatrix  $F_1$  für die Helligkeit des hervorgehobenen Pixels

$$h' = \frac{1}{16} \cdot (1 \cdot 0,37 + 2 \cdot 0,94 + 1 \cdot 0,89 + 2 \cdot 0,18 + 4 \cdot 0,60 + 2 \cdot 0,89 + 1 \cdot 0,05 + 2 \cdot 0,26 + 1 \cdot 0,70) = 0,56.$$

Da die Helligkeitswerte von Pixeln durch Faltungsmatrizen, welche nur aus positiven Komponenten bestehen, an die Werte benachbarter Pixel angenähert werden, bewirken derartige Matrizen eine *Weichzeichnung* von Bildern. Demgegenüber erfolgt durch Matrizen, die in Umgebungen der Pixel, auf die sie angewendet werden, negative Elemente enthalten, die Verstärkung von Helligkeitsunterschieden und somit eine *lokale Kontrasterhöhung* bzw. *Kanten hervorhebung* (siehe u. a. [122], S. 103–130; [124], S. 130–139 und [187], S. 53–68).

- Häufig sind *Transformationen* aus der durch Bildmatrizen gegebenen ortsabhängigen Helligkeits- bzw. Farbbeschreibung *in den Frequenzbereich* notwendig, indem Helligkeitsverläufe durch Summen von Sinus- bzw. Kosinusfunktionen beschrieben werden. Dieser Ansatz geht auf die Signalverarbeitung zurück, da Bilder als zweidimensionale Signale aufgefasst werden können. Für die Transformationen aus dem Orts- in den Frequenzbereich kommen diskrete Fouriertransformationen zur Anwendung ([122], S. 40–62; [124], S. 156–199; [187], S. 118–126 und [258], S. 262–289). Die Bildrepräsentation im Frequenzbereich ermöglicht komplexe Bildveränderungen. So ist es z. B. durch die Unterdrückung höherfrequenter Signalanteile (d. h. durch die Verkleinerung der höherfrequenten Koeffizienten der Fourierreihen) möglich, das Rauschen eines Bildes zu vermindern, wobei allerdings auch feinere Bilddetails verloren gehen können. Gezielte Eingriffe sind möglich, wenn das Rauschspektrum des Bildaufnahmegerätes bekannt ist. Eine weitere Anwendung der Transformationen in den Frequenzbereich ist die Bildkomprimierung mithilfe der Wavelet-Transformation (siehe [124], S. 170–180).

Außer für Bildmanipulationen kommen Nachbarschaftsoperationen und Fouriertransformationen auch in der (derzeit stark an Bedeutung gewinnenden) Bilderkennung (Merkmalsextraktion) zum Einsatz. Derartige Verfahren können z. B. genutzt werden, um von einer im Kraftwagen montierten Kamera aufgenommene Bilder daraufhin zu untersuchen, welche Verkehrsschilder am Straßenrand stehen. Auch die Vektorisierung von Bildern ist der Merkmalsextraktion zuzuordnen. Dabei werden durch Kantendetektion in Bildern auftretende Strecken sowie Kurven erkannt und analytisch beschrieben.

## Vektorgrafik – Objektorientierte Grafikdarstellung

Der Ansatz der Vektorgrafik unterscheidet sich von dem der Rastergrafik insofern grundsätzlich, als hier einzelne geometrische Objekte der Grafik beschrieben werden. Eine Strecke wird z. B. durch die Koordinaten ihrer beiden Endpunkte, ein Kreis durch die Koordinaten seines Mittelpunktes und den Radius festgelegt. Neben diesen geometrischen Bestimmungsgrößen werden Objekten in der Vektorgrafik vielfältige weitere Attribute wie Linienstärke und -farbe, Füllung und Transparenz zugeordnet. Sowohl die Positionen als auch die Eigenschaften von Elementen bleiben in Vektorgrafiken veränderbar – es ist also beispielsweise jederzeit möglich, die Lage eines Kreises, seinen Radius, die Dicke, mit der die Peripherie des Kreises gezeichnet wird, sowie die Farbe und Transparenz, mit der das Kreisinnere gefüllt ist, zu verändern. Gespeichert werden in der Vektorgrafik keine Bildpunkte, sondern die Koordinaten von bestimmenden Punkten sowie andere Bestimmungsgrößen der Objekte. Dies hat neben der Möglichkeit, Objekte nachträglich zu verändern, zur Folge, dass die Dateigrößen von Vektorgrafiken oft erheblich kleiner sind als die von Rasterbildern.

Bei der praktischen Arbeit mit Vektorgrafik-Software<sup>10</sup> werden geometrische Objekte durch Ziehen mit der Maus erstellt; die Bestimmungspunkte und -größen können auch mausgesteuert verändert werden. Außerdem ist es möglich, durch Zahleneingaben Positionen und Bestimmungsgrößen zu verändern sowie Transformationen durchzuführen. Als Beispiel zeigt Abb. 2.3 eine in Corel Designer gezeichnete Ellipse und das zugehörige Dialogfeld, in dem die Koordinaten ihres Mittelpunktes, die Längen der Hauptachsen sowie der Winkel einer Hauptachse zur Horizontalen verändert werden können.

Da sowohl im Grafikdesign, in dem Vektorgrafikprogramme hauptsächlich zum Einsatz kommen, als auch bei der computergestützten Konstruktion geometrische Grundfiguren wie Strecken, Ellipsen und Polygone in den meisten Fällen keine hinreichenden Gestaltungsmöglichkeiten bieten, sind so genannte Freiformkurven zu den wichtigsten geometrischen Objekten der Vektorgrafik geworden. Dabei handelt es sich um Kurven, die durch Stütz- oder Kontrollpunkte und Tangenten bestimmt werden. Die größte Bedeutung haben dabei Bézier- sowie B-Splinekurven erlangt. Abb. 2.4 zeigt eine Bézierkurve mit ihren Stützpunkten und Steuertangenten. Durch Verschieben der Stützpunkte sowie

<sup>10</sup>Die bekanntesten Programme dieser Kategorie sind Corel Draw und Corel Designer, Macromedia Freehand sowie Adobe Illustrator. Alle hier getroffenen Aussagen gelten für jedes dieser Programme.

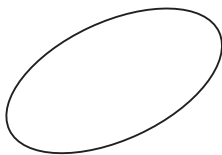


Abbildung 2.3: Geometrische Bestimmungsgrößen einer Ellipse in Corel Designer

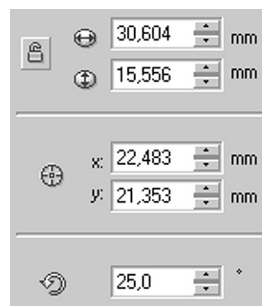


Abbildung 2.4: Bézierkurve

der „Anfasspunkte“ der Steuertangenten kann die Form der Kurve verändert werden. Die mathematische Beschreibung der Bézier- und B-Spline-Kurven sowie vergleichbarer Flächen des Raumes ist Gegenstand der Abschnitte 2.3.6 und 2.3.8.

Nach dem Prinzip der Vektorgrafik arbeiten auch Softwarepakete für computergestütztes Konstruieren (CAD, siehe Abschnitt 2.7.4). In diesem Anwendungsbereich ist eine exakte Steuerung der Positionen und Maße einzelner Objekte besonders bedeutsam. Hinsichtlich der objektorientierten Grafikrepräsentation ist auch Dynamische Geometriesoftware (DGS) in den Bereich der Vektorgrafik einzuordnen. Die Besonderheit dieser Kategorie von Software besteht darin, dass die Koordinaten der bestimmenden Punkte geometrischer Objekte dynamisch von anderen Objekten abhängen können.<sup>11</sup>

Da fast alle heute verwendeten Ausgabegeräte (Bildschirme und Drucker) Punktraster ausgeben,<sup>12</sup> müssen Vektorgrafiken vor der Darstellung auf dem Bildschirm oder dem Ausdruck *gerastert*, also letztendlich in Pixelgrafiken umgewandelt werden. Diese Rasterung geschieht meist in der Ansteuerungssoftware der betreffenden Geräte (Drucker- und Grafiktreiber) und bleibt vor den Anwendern verborgen. Bei Vektorgrafiken wird die Rasterung jeweils mit der höchsten Auflösung vorgenommen, die das jeweilige Ausgabegerät darstellen kann.<sup>13</sup> So können Vektorgrafiken prinzipbedingt auf jedem Ausgabegerät mit der maximalen, nur durch das Gerät begrenzten, Qualität und Detailtreue dargestellt sowie beliebig skaliert werden. Hingegen liegen Bitmapgrafiken bereits als Punktmatrizen mit einer gegebenen Auflösung vor; ihre Vergrößerung kann deshalb keine detailliertere Darstellung von Bildinhalten ermöglichen und führt lediglich zu einer Vergrößerung der dann auf eine größere Zahl von Pixeln verteilten Bildinformationen. Abbildung 2.5 illustriert diesen Unterschied anhand einer Vektorgrafik (a, b) und einer Rastergrafik mit der Auflösung 150 dpi (c) in jeweils einfacher sowie zweifacher Vergrößerung. Die Rastergrafik wurde mit zwei verschiedenen Verfahren vergrößert: einfache Pixelwiederholung (d) und Interpolation „hinzugerechneter“ Pixel aus benachbarten Pixeln (e).

<sup>11</sup>Zwar ist es auch in einigen Vektorgrafik- sowie in allen CAD-Systemen möglich, Grafikobjekte an anderen Objekten auszurichten. So ermöglichen es Fangfunktionen beispielsweise, beim Zeichnen einer Strecke einen Endpunkt in den Schnittpunkt zweier anderer Strecken einzurasten. Werden diese jedoch verändert, so wirkt sich das nicht auf die zuletzt konstruierte Strecke aus. DGS erlauben hingegen die Konstruktion dauerhaft abhängiger Objekte. Koordinaten von bestimmenden Punkten werden dabei nicht als feste Werte gespeichert, sondern bei Veränderung der „Elternfiguren“ neu berechnet.

<sup>12</sup>Eine Ausnahme bilden dabei lediglich Plotter, bei denen Stifte Strecken und Kurven zeichnen.

<sup>13</sup>Die Auflösung üblicher Monitore liegt zwischen 72 und 96 dpi (dots per inch, Punkte je Zoll), die von Druckern ist wesentlich höher (300-1200 dpi).

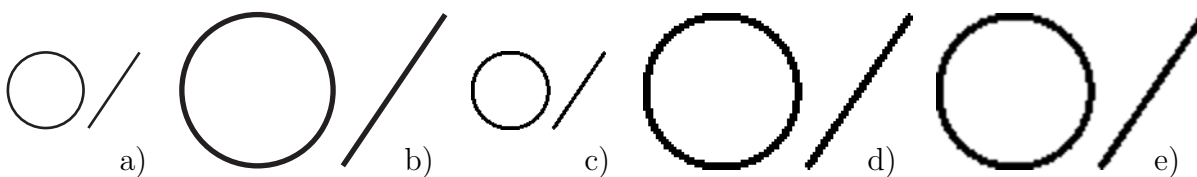


Abbildung 2.5: Vektorgrafik und Bitmap (150 dpi) in ein- und zweifacher Vergrößerung

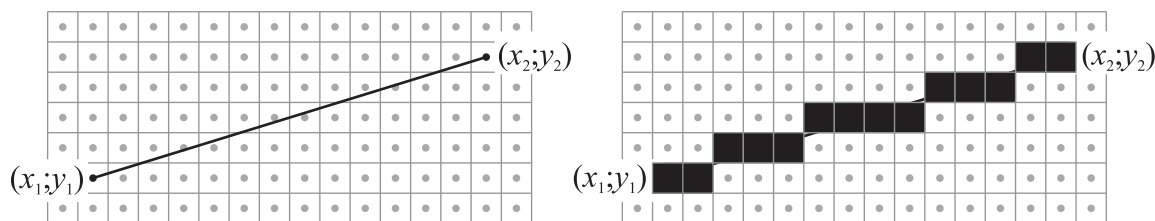


Abbildung 2.6: Rasterung einer Strecke

In der Anfangsphase der computergestützten Grafikbearbeitung und -darstellung wurde der *Rasterung* grafischer Objekte (auch als *Rasterkonversion* oder *Scan Conversion* bezeichnet) große Aufmerksamkeit zuteil. Es war notwendig, Algorithmen zu entwickeln, die bei der Umsetzung von analytisch gegebenen Grafikobjekten in Pixelraster einen guten Kompromiss zwischen der Qualität der Darstellung und der benötigten Rechenzeit ermöglichen.<sup>14</sup> Um beispielsweise eine Strecke auf dem Bildschirm anzuzeigen, müssen zunächst die Koordinaten ihrer Endpunkte bezüglich des Koordinatensystems, mit dem das Grafikprogramm arbeitet, durch Skalieren und Runden in Bildschirmkoordinaten überführt werden. Diese sind natürliche Zahlen, da jeder Bildschirm über eine endliche Zahl von Pixeln verfügt (gebräuchlich sind z. B. 1280 Pixel in horizontaler und 1024 Pixel in vertikaler Richtung). Aus den Bildschirmkoordinaten der Endpunkte ist dann zu ermitteln, durch welche Quadrate des Bildschirmrasters die Strecke verläuft; die entsprechenden Pixel stellen die Strecke dann auf dem Bildschirm dar.

Sind  $(x_1; y_1)$  und  $(x_2; y_2)$  mit  $x_1, y_1, x_2, y_2 \in \mathbb{N}$  und  $x_1 < x_2$  die Koordinaten der Endpunkte einer Strecke, so besteht eine einfache Möglichkeit der Rasterung dieser Strecke darin, für alle natürlichen Zahlen  $x \in [x_1; x_2]$  die zugehörigen Werte für  $y$  anhand der Zweipunktgleichung

$$y = \frac{y_2 - y_1}{x_2 - x_1} \cdot (x - x_1) + y_1$$

der Strecke zu bestimmen und auf ganzzahlige Werte zu runden. Die so ermittelten ganzzahligen Koordinatenpaare geben die Pixel an, durch welche die Strecke dargestellt wird (siehe Abb. 2.6). Da der Anstieg einer Strecke i. Allg. keine ganze Zahl ist, müssen für die Bestimmung der Position jedes Pixels Gleitkommaoperationen (Berechnungen mit rationalen Zahlen) durchgeführt werden. Da derartige Berechnungen vergleichsweise viel Rechenzeit benötigen, wurden Algorithmen entwickelt, die es ermöglichen, Strecken und andere geometrische Figuren durch die alleinige Anwendung von Ganzzahloperationen zu rastern. Am häufigsten kommen dafür die inkrementellen Algorithmen von BRESENHAM zur Anwendung (siehe u. a. [11], S. 57–63; [57], S. 60–63; [79], S. 25–40; [236], S. 173–175 und [282], S. 45–66). Deren Prinzip besteht darin, schrittweise jeweils aus zwei für die Darstellung eines geometrischen Objekts in Frage kommenden Pixeln dasjenige auszuwählen, welches näher an dem Objekt liegt. Die Entscheidung, für welches zweier Pixel die Abweichung zu der „idealen“ Lage eines

<sup>14</sup>Bis Mitte der neunziger Jahre verursachte die für die Rasterung benötigte Zeit spürbare Verzögerungen im Bildschirmaufbau bei der Darstellung komplexerer Vektorgrafiken. Aufgrund der erheblich gestiegenen Rechenleistungen sind derartige Verzögerungen nur noch in Einzelfällen feststellbar.

Punktes des Objekts geringer ist, kann auf Ganzzahloperationen zurückgeführt werden; die ermittelten Werte lassen sich für die Zuordnung des jeweils nächsten Pixels nutzen. Sowohl die Rasterung von Strecken als auch die von Kreisen und Ellipsen sind auf diese Weise sehr schnell ausführbar.

Wie in Abb. 2.6 zu erkennen ist, führt die Rasterung prinzipbedingt zu einem Treppeneffekt, auch als *Aliasing* bezeichnet, der bei der Darstellung auf Bildschirmen (mit Abständen benachbarter Pixel von ca. 0,3 mm) als störend empfunden werden kann. Zur Abschwächung dieses Effekts dienen Kantenglättungs- bzw. *Anti-Aliasing-Verfahren* (siehe z. B. [31], S. 121–126 und [282], S. 68–72). Das gebräuchlichste dieser Verfahren ersetzt harte Kanten durch mehrstufige Helligkeitsübergänge. Um den Treppeneffekt bei der Rasterung einer Strecke zu vermindern, lässt sich anstelle der Strecke ein Rechteck mit der Breite eines Pixels betrachten und jedes Pixel in Teilquadrate unterteilen. Im einfachsten Falle erfolgt eine Unterteilung in 4 Quadrate (siehe Abb. 2.7 a). Die beiden zu der ursprünglichen Strecke parallelen Seiten des Rechtecks werden nun bezüglich des verfeinerten Rasters gerastert (Abb. 2.7 b). Da die dabei berechneten Pixel auf dem Bildausgabegerät mit dem ursprünglichen groben Raster nicht darstellbar sind, werden den größeren Quadraten Helligkeits- (Grau-)werte zugeordnet: 100% (reines Weiß), wenn auf einem Quadrat kein schwarzes Teilquadrat liegt, 75% Helligkeit bei einem, 50% bei zwei und 25% bei drei schwarzen Teilquadraten. Ist ein Pixel von vier schwarzen Teilquadraten bedeckt, so wird es völlig schwarz, also mit 0% Helligkeit dargestellt. Abb. 2.7 c) zeigt die so erzeugte Rasterung einer Strecke durch 4 Graustufen; in Abb. 2.7 d) sind oben die einfach und unten die mit Anti-Aliasing gerasterte Strecke jeweils in zehnfach kleinerem Maßstab abgebildet.

Auch für die 3D-Computergrafik ist Anti-Aliasing von großer Bedeutung für die Qualität der erzeugten Grafiken, wobei dort wesentlich komplexere und hinsichtlich der benötigten Rechenzeit aufwändigere Verfahren zum Einsatz kommen (siehe 2.5.7).

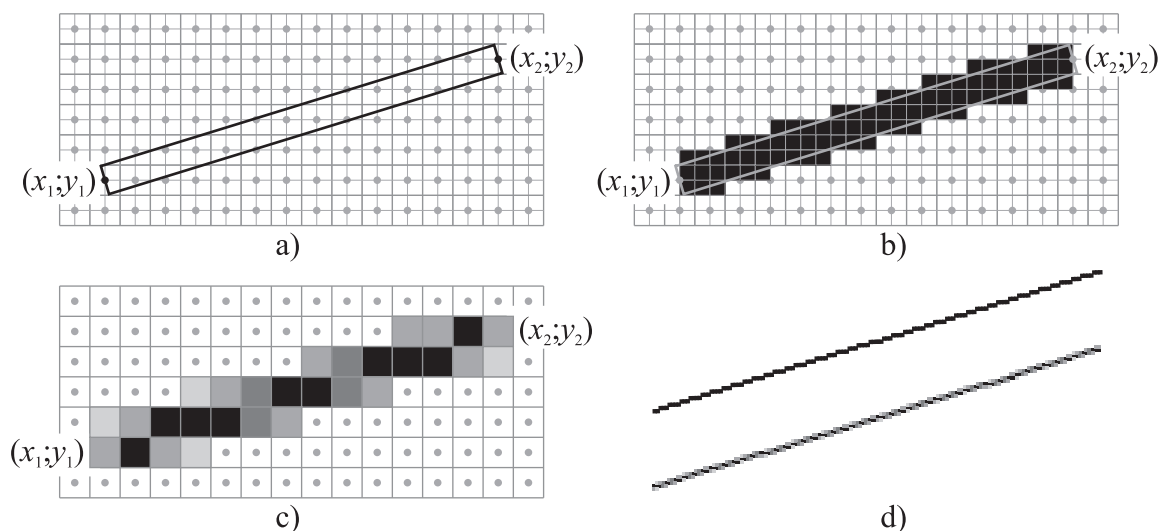


Abbildung 2.7: Anti-Aliasing

### 2.1.2 Licht und Farben

#### Physikalische Grundlagen; Modelle für das Licht

In der Physik sind drei Modelle für die Beschreibung des Lichtes gebräuchlich:<sup>15</sup>

1. In der *Strahlenoptik* wird die Ausbreitung von Lichtstrahlen durch einfache geometrische Modelle beschrieben. Damit können optische Erscheinungen, die farbunabhängig sind, wie Reflexion und Brechung (Refraktion) erfasst werden.
2. Der Charakter des Lichtes als elektromagnetischer Wellen liegt der *Wellenoptik* zugrunde. Neben der Erklärung von Interferenz, Beugung (Diffraktion) und Polarisation ermöglicht es dieses Modell auch, Farben von Licht durch Wellenlängen elektromagnetischer Wellen zu beschreiben und die unterschiedliche (wellenlängenabhängige) Ausbreitungsgeschwindigkeit von Lichtstrahlen verschiedener Farben (Dispersion) zu berücksichtigen.
3. Der *Quantenoptik* liegt das Teilchenmodell zugrunde, in dem Lichtstrahlen als Partikel (Photonen) aufgefasst werden. Mithilfe dieses Modells lassen sich u. a. der photoelektrische Effekt und die Emission eines schwarzen Körpers erklären.

Für Bildberechnungen in der 3D-Computergrafik kommt weitgehend das Strahlenmodell des Lichtes zum Einsatz, welches ausreicht, um den Verlauf von Lichtstrahlen zu berechnen. Dadurch wird die Bildberechnung auf die Geometrie des Raumes zurückgeführt. Da sich die geometrische Optik nur mit dem Verlauf von Lichtstrahlen, nicht aber mit ihrer Farbe (die von der Wellenlänge abhängt) befasst, werden bei der Erzeugung farbiger Grafiken Helligkeitsberechnungen für mehrere Grundfarben durchgeführt, wofür die im Folgenden vorgestellten vektoriellen Farbmodelle zur Anwendung kommen. Diesen Modellen liegt die Wellenauffassung des Lichtes zugrunde, die somit mittelbar ebenfalls eine Grundlage der Computergrafik bildet. Spezielle Berechnungsmethoden in der Computergrafik greifen auch auf das Photonenmodell zurück, um Effekte der indirekten Reflexion und der globalen Beleuchtung zu simulieren (siehe Abschnitt 2.5.6).

#### Psychophysische Grundlagen der Beschreibung von Farben durch Vektoren

Als Licht werden nach der von MAXWELL 1864 aufgestellten elektromagnetischen Lichttheorie elektromagnetische Wellen mit einer Wellenlänge  $\lambda$  von ca. 380–780 Nanometern aufgefasst. In diesem Bereich wird die von den Wellen transportierte Energie vom menschlichen Auge wahrgenommen. Die Farbwahrnehmung hängt von der Wellenlänge der Lichtwellen ab und reicht von violett (für die kürzesten sichtbaren Wellenlängen) über blau, grün und gelb bis rot (für die längsten sichtbaren Wellenlängen), siehe Abb. B.1 auf S. 349 in Anhang B. Aus Abb. 2.8 geht der Verlauf der Lichtempfindlichkeit des Auges in Abhängigkeit von der Wellenlänge hervor.

Nach der *Tristimulustheorie* befinden sich auf der Netzhaut (Retina) des menschlichen Auges drei Arten von Sensoren (Synapsen) mit unterschiedlichen frequenz- bzw.

---

<sup>15</sup>Eine ausführliche Darstellung der physikalischen Grundlagen dieser Modelle würde weit über den Rahmen dieser Arbeit hinausgehen. Es sei hierzu auf Lehrbücher der Optik (wie z. B. [198]) verwiesen.



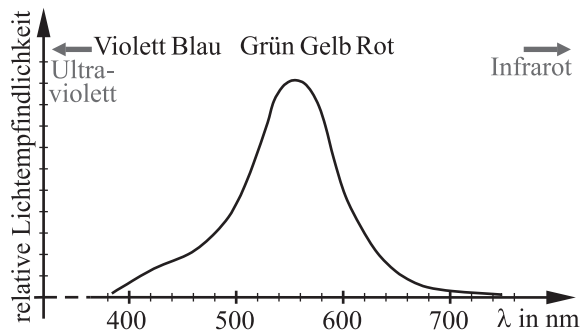
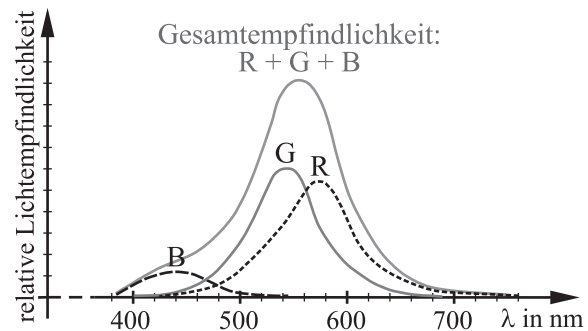
Abbildung 2.8: Empfindlichkeit des Auges in Abhängigkeit von der Wellenlänge  $\lambda$ 

Abbildung 2.9: Empfindlichkeiten der drei Arten von Synapsen

wellenlängenabhängigen Empfindlichkeiten; Abb. 2.9 gibt die Empfindlichkeitsverläufe wieder, wobei die drei Arten von Synapsen nach den ihren Empfindlichkeitsmaxima zuzuordnenden Farben mit B (Blau), G (Grün) und R (Rot) bezeichnet sind. Es geht daraus hervor, dass das Auge auf bläuliche Farben weniger sensibel reagiert als auf grüne, gelbe und hellrote Farbtöne. Da in der Natur nur sehr selten monochromatisches Licht (Licht einer einzigen Wellenlänge) auftritt, resultieren Farbempfindungen meist aus der Wahrnehmung eines Gemisches unterschiedlicher Wellenlängen. Einem solchen Gemisch wird durch das menschliche Gehirn in Abhängigkeit von der Anregung der drei Synapsenarten ein Farbeindruck zugeordnet.<sup>16</sup>

Aus der Tristimulustheorie ergibt sich die Möglichkeit, durch die Addition von Licht dreier Primärfarben beliebige Farbwahrnehmungen zu erzeugen, worauf die elektronische Bildwiedergabetechnik und die in der Computergrafik verwendeten Farbmodelle beruhen. Bildwiedergabegeräte wie Bildröhren und Flachbildschirme verfügen über Bildmatrizen mit jeweils einem roten, einem grünen und einem blauen Lichtemitter in jedem Matrixpunkt. Da diese sehr eng beieinander liegen, kann das Auge die einzelnen Farbkomponenten nicht mehr erkennen und „addiert“ ihre Helligkeitswerte. So wird z. B. bei maximaler Helligkeit aller drei Komponenten die Farbe Weiß wahrgenommen.

1931 verabschiedete die CIE (Commission Internationale de l'Eclairage) einen Standard für ein Modell, in dem jede Farbe als gewichtete Summe (Linearkombination) dreier Grundfarben dargestellt werden kann. Die Charakterisierung dieser (synthetischen) Grundfarben X, Y und Z erfolgt durch Energieverteilungskurven (vgl. u. a. [64], S. 32–35 und [79], S. 459–463). Da reale Geräte diese Farben nicht erzeugen können, kommt das CIE-XYZ-Modell in der Technik nicht direkt zum Einsatz.<sup>17</sup> Das in der Computergrafik dominierende RGB-Modell beruht jedoch auf demselben Ansatz der Darstellung von Farben als Linearkombinationen dreier Grundfarben.

<sup>16</sup>Diese Zuordnung ist natürlich nicht eineindeutig. Ein und derselbe Farbeindruck kann stets durch unendlich viele spektrale Verteilungen von Lichtwellen erzeugt werden.

<sup>17</sup>Die Wahl der Grundfarben für das CIE-XYZ-Modell erfolgte mit dem Ziel, den gesamten für das menschliche Auge erfassbaren Farbraum (Farbgamut) zu beschreiben und orientierte sich nicht daran, welche Farben technische Geräte tatsächlich darstellen können. Der durch fotografische und technische Geräte darstellbare Farbraum umfasst stets nur eine Teilmenge des CIE-XYZ-Farbraumes.

### Das additive Farbmodell (RGB-Modell)

Im RGB-Farbmodell werden alle darstellbaren Farben aus den drei Grundfarben Rot, Grün und Blau gemischt (addiert), denen folgende Wellenlängen entsprechen:

$$\lambda_R \approx 650 \text{ nm}, \quad \lambda_G \approx 530 \text{ nm}, \quad \lambda_B \approx 460 \text{ nm}.$$

Jede Farbe lässt sich im RGB-Modell durch ein Tripel  $\begin{pmatrix} r \\ g \\ b \end{pmatrix}$  beschreiben.<sup>18</sup> Da das

menschliche Auge nicht beliebig kleine Farbabweichungen unterscheiden kann, genügt es i. Allg., für jede Komponente 256 unterschiedliche Werte zuzulassen.<sup>19</sup> Die Komponenten des Farbtripels werden meist als natürliche Zahlen von 0 bis 255 oder als rationale Zahlen des Intervalls  $[0, 1]$  dargestellt. Im Folgenden wird letztere Darstellung genutzt. Es können beliebige Werte von 0 bis 1 für jede Farbkomponente eingegeben werden, wobei Grafikprogramme diese auf ganzzahlige Vielfache von  $\frac{1}{255}$  runden.

Für die Menge aller durch das RGB-Modell darstellbaren Farben ist die Bezeichnung „RGB-Farbraum“ gebräuchlich, obwohl es sich hierbei aufgrund der Beschränkung  $r, g, b \in [0, 1]$  nicht um einen Vektorraum handelt.<sup>20</sup> Mit der Vektoraddition und der Multiplikation von Vektoren mit Skalaren vergleichbare Operationen können dennoch sinnvoll auf Farben angewendet werden. Wird die Addition zweier Farbvektoren durch

$$\begin{pmatrix} r_1 \\ g_1 \\ b_1 \end{pmatrix} + \begin{pmatrix} r_2 \\ g_2 \\ b_2 \end{pmatrix} := \begin{pmatrix} \min(r_1 + r_2, 1) \\ \min(g_1 + g_2, 1) \\ \min(b_1 + b_2, 1) \end{pmatrix}$$

definiert, so entspricht das Ergebnis dem Resultat, das sich bei der Beleuchtung einer Fläche mit zwei Scheinwerfern der Farbwerte  $(r_1; g_1; b_1)^T$  und  $(r_2; g_2; b_2)^T$  ergibt.

Die Addition von Farben ist in einem Bildbearbeitungsprogramm, das die Anordnung von Pixeln auf Ebenen unterstützt, nachvollziehbar. Werden z. B. in Adobe Photoshop drei Kreise mit den Grundfarben Rot, Grün und Blau auf jeweils eine Ebene über einer schwarzen Hintergrundebene gelegt und wird für diese Ebenen der Modus „Aufhellen“ (Addition der  $r$ -,  $g$ - und  $b$ -Komponenten übereinander liegender Pixel) gewählt, so ergibt sich ein Bild wie in Abb. 2.10 bzw. in der farbigen Abb. B.2 auf S. 349.<sup>21</sup> Pixel, die im Durchschnitt aller drei Kreise liegen, nehmen die Farbe Weiß an:

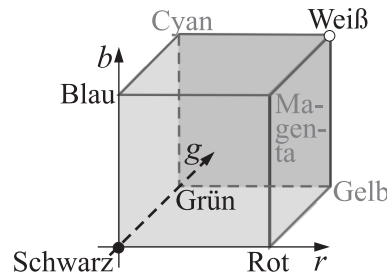
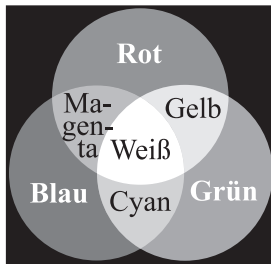
$$R + G + B = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = W.$$

<sup>18</sup>Komponenten von Farben werden mit Klein-, Farben selbst mit Großbuchstaben bezeichnet.

<sup>19</sup>Die Festlegung auf 256 Werte erfolgte, weil dadurch genau 8 Bit an Informationen für die Beschreibung jeder Komponente notwendig sind ( $2^8 = 256$ ). Häufig wird auch die Hexadezimalschreibweise genutzt, bei der jede „Ziffer“ 16 Werte (0...9, A...F) annehmen kann. Für die Darstellung einer Farbe genügen dann 2 Stellen. Für Anwendungen mit sehr hohen Qualitätsansprüchen werden 16 Bit für die Beschreibung jeder Farbkomponente genutzt, damit sind 65536 verschiedene Abstufungen möglich.

<sup>20</sup>In der Literatur sind dennoch die Bezeichnungen „Farbvektor“ bzw. „RGB-Vektor“ gebräuchlich.

<sup>21</sup>Mit der Datei `Farbaddition.psd` (siehe Anhang D, S. 365) kann die Farbaddition in Adobe Photoshop oder dazu kompatibler Bildbearbeitungssoftware nachvollzogen werden.

Abbildung 2.10:  
RGB-MischungAbbildung 2.11:  
RGB-Würfel

Als Summe von Rot und Grün ergibt sich die Farbe Gelb (Abkürzung Y von Yellow), als Summe von Rot und Blau Magenta (M) sowie von Grün und Blau Cyan (C):

$$R + G = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} = Y, \quad R + B = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = M, \quad G + B = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} = C.$$

Die Summe zweier Grundfarben ist dabei jeweils die Komplementärfarbe der nicht an der Summenbildung beteiligten Grundfarbe; somit sind Gelb und Blau, Rot und Cyan sowie Grün und Magenta jeweils Paare von Komplementärfarben.

Da die Menge aller Farbvektoren  $\begin{pmatrix} r \\ g \\ b \end{pmatrix}$  eine Teilmenge von  $\mathbb{R}^3$  (mit  $0 \leq r, g, b \leq 1$ )

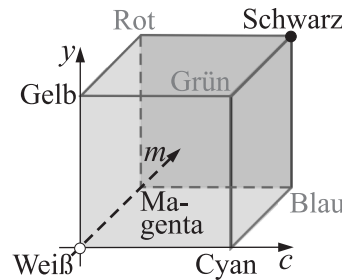
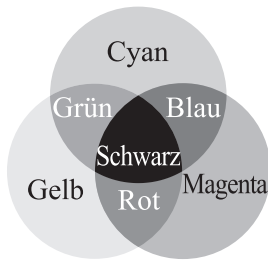
ist, kann sie als Würfel im Anschauungsraum dargestellt werden; es ergibt sich der *RGB-Würfel*.<sup>22</sup> Dabei entsprechen komplementären Farben jeweils gegenüberliegende Punkte des Würfels (Eckpunkte, die auf einer Raumdiagonalen liegen). Die additive Farbmischung kann sinnvoll auf die Addition von Pfeilen des Anschauungsraumes zurückgeführt werden, wenn die resultierenden Pfeile innerhalb des RGB-Würfels liegen.

### Das subtraktive Farbmodell (CMY- bzw. CMYK-Modell)

Der Koordinatenursprung im RGB-Modell entspricht dem Nichtvorhandensein von Licht und somit dem Ruhezustand elektronischer Bildwiedergabegeräte. Demgegenüber verhält sich durch weißes Licht beleuchtetes Papier umgekehrt: In seinem Ausgangszustand, also unbedruckt, reflektiert ideales Papier die gesamte auftreffende Helligkeit; durch das Aufbringen von Farbpigmenten erfolgt eine Subtraktion von Helligkeitsanteilen. Werden Pigmente aufgetragen, die bestimmte Wellenlängen absorbieren, nimmt Papier die Farbe des „Restlichtes“ an, bei der es sich um die Komplementärfarbe der absorbierten Farbe handelt. Somit wird die Farbwiedergabe im Druck durch das subtraktive CMY-Modell mit den Grundfarben Cyan, Magenta und Gelb – den Komplementärfarben der Grundfarben des RGB-Modells – beschrieben. Abb. 2.12 sowie die farbige Abb. B.4 auf S. 349 zeigen die subtraktive Farbmischung dieser Farben.

Auch im CMY-Modell lässt sich die Menge aller Farben als Würfel im Anschauungsraum darstellen. In diesem CMY-Würfel (siehe Abb. 2.13) entspricht der Koordinatenursprung dem Weißpunkt – für diesen Punkt erfolgt kein Farbauftrag. Die Koordina-

<sup>22</sup>Siehe Abb. 2.11 bzw. die Farabbildung auf S. 349 sowie das Video *Farbwuerfel-Video.mpg* (vgl. Anhang D, S. 365), in dem der RGB-Würfel aus verschiedenen Richtungen sichtbar ist.

Abbildung 2.12:  
CMY-MischungAbbildung 2.13:  
CMY-Würfel

tenachsen geben die Farbaufräge für die drei Grundfarben C, M und Y an. Zwischen den RGB- und den CMY-Koordinaten einer Farbe besteht der Zusammenhang

$$\begin{pmatrix} c \\ m \\ y \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} - \begin{pmatrix} r \\ g \\ b \end{pmatrix}.$$

In der Praxis ist die Transformation vom RGB- in den CMY-Farbraum komplizierter, da weder Papier noch Farbstoffe ein ideales Verhalten aufweisen. Insbesondere kann eine reale Druckmaschine aus den drei Grundfarben kein wirkliches Schwarz erzeugen. Als vierte Farbe kommt daher Schwarz zur Anwendung – das CMY-Modell erfährt eine Erweiterung zum CMYK-Modell (K für Black). Die Farbtransformation aus dem RGB- in den CMYK-Raum wird als *Farbseparation* bezeichnet, siehe z. B. [38] und [190].

### Das HSB (bzw. HSV-) und das HLS-Modell

Das RGB- und das CMY(K)-Modell sind Farbmodelle, die sich an den Eigenschaften von Bildwiedergabegeräten sowie, im Falle des RGB-Modells, an der menschlichen Farbwahrnehmung orientieren. Demgegenüber wurden Farbmodelle entwickelt, die besonders gut für Bildkomposition und Design genutzt werden können.<sup>23</sup> Jede Farbe lässt sich durch die folgenden 3 Eigenschaften beschreiben:

- Der *Farbton* (H) ergibt sich aus der im Spektrum der betreffenden Farbe dominierenden Wellenlänge und kann durch Begriffe wie Rot, Grün, Orange usw. beschrieben werden. Die Menge aller Farbtöne wird oft mithilfe des Farbkreises (siehe Abb. B.5 auf S. 349 in Anhang B) dargestellt.
- Die *Sättigung* (S) gibt an, wie „rein“ eine Farbe ist. S ist groß, wenn innerhalb des Gesamtspektrums des betrachteten Lichtes eine einzelne Wellenlänge oder ein schmaler Bereich von Wellenlängen stark dominiert. Treten im Spektrum viele sehr unterschiedliche Wellenlängen mit annähernd gleicher Intensität auf, so verfügt die Farbe über eine geringe Sättigung und wirkt blass bzw. gräulich.
- Die *Helligkeit* (B bzw. V) ergibt sich aus den Intensitäten der Lichtwellen, aus denen sich eine Farbe zusammensetzt.

<sup>23</sup>Dazu gehören vor allem das HSB- (bzw. HSV-) sowie das HLS-Modell. H steht für Hue (Farbton), S für Saturation (Farbsättigung), B für Brightness (Helligkeit, diese wird auch oft mit V – Value bezeichnet) sowie L für Lightness. Da das HSB- und das HLS-Modell einem vergleichbaren Ansatz folgen und sich im Aufbau ähnlich sind, das HSB-Modell jedoch weitaus häufiger in Bildbearbeitungs- und Grafiksoftware zur Verfügung steht, wird im Folgenden auf dieses Modell eingegangen.

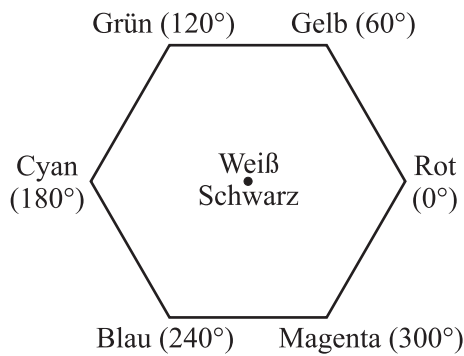


Abbildung 2.14: Farbsechseck

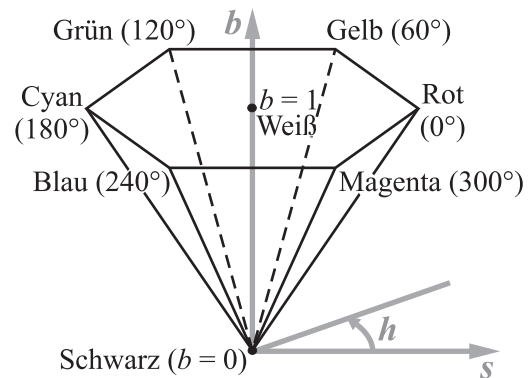


Abbildung 2.15: HSB-Pyramide

Das HSB-Modell unterstützt gut die Auswahl harmonisch wirkender Farben. Die wichtigste Regel dazu besteht darin, Farben mit annähernd gleicher Sättigung zu verwenden, da Farben mit großen Sättigungsunterschieden als zueinander unpassend empfunden werden. In vielen Fällen ist ferner die Auswahl ähnlicher Helligkeitswerte sinnvoll und für zusammengehörige, in ihrer Funktion korrespondierende, Grafikelemente empfiehlt es sich, komplementäre Farbtöne zu verwenden (vgl. [33], S. 135).

Um vom RGB- zum HSB-Modell zu gelangen, wird der RGB-Würfel (siehe Abb. 2.11 auf S. 77) parallel zur Schwarz-Weiß-Diagonalen auf die zu dieser Diagonalen im Weißpunkt orthogonale Ebene projiziert. Dabei entsteht ein regelmäßiges Sechseck mit den Eckpunkten Rot, Gelb, Grün, Cyan, Blau und Magenta (Abb. 2.14). Jedem Farbton außer Schwarz, Weiß und reinen Grautönen ist in diesem Sechseck ein Winkel  $h$  im Intervall  $[0; 360^\circ)$  zugeordnet. Diese Zuordnung liegt auch dem *Farbkreis* (siehe Abb. B.5 auf S. 349) zugrunde. Der Abstand einer Farbe vom Mittelpunkt ist das Maß für ihre Sättigung. Durch Betrachtung einer regelmäßigen Pyramide mit dem Farbsechseck als Basis lässt sich ein Koordinatensystem mit der Spitze der Pyramide als Ursprung und ihrer Achse als einer Koordinatenachse (Helligkeits- bzw.  $b$ -Achse) einführen (Abb. 2.15). Einer Farbe mit den RGB-Koordinaten  $(r_F; g_F; b_F)$  wird als Helligkeit  $b$  das Maximum der Farbwerte  $r_F$ ,  $g_F$  und  $b_F$  zugeordnet. Wird eine zweite Koordinatenachse (Sättigungs- bzw.  $s$ -Achse) durch die Projektion des Weiß-Rot-Strahls auf die durch den Koordinatenursprung verlaufende, zur  $b$ -Achse senkrechte Ebene festgelegt, so können Farben durch die Zylinderkoordinaten  $(s; h; b)$  beschrieben werden.

### Weitere Farbmodelle

Neben den beschriebenen Farbmodellen existiert eine Reihe weiterer Modelle, die jedoch in der Computergrafik eine geringere Bedeutung haben. Dazu zählen das im Farbfernsehen nach dem NTSC-Standard verwendete YIQ-Farbmodell (siehe [79], S. 466 und [282], S. 294) sowie das CIE-LAB-Modell (siehe [210]). In jedem dieser Modelle entspricht eine Komponente (Y bzw. L) der Helligkeit, die beiden anderen Komponenten geben die Lage der beschriebenen Farbe auf jeweils zwei Farbskalen an.

## 2.2 Überblick über die Funktionsweise der 3D-Computergrafik

In diesem Abschnitt soll eine Bestimmung des Begriffs der dreidimensionalen Computergrafik anhand ihrer grundlegenden Wirkungsweise vorgenommen werden. Der Begriff „dreidimensional“ bezieht sich nicht auf das Ergebnis – auch dreidimensionale computergrafische Darstellungen werden fast immer auf zweidimensionalen Medien (Papier oder elektronische Bildwiedergabegeräte) betrachtet – sondern auf die Erzeugung computergrafischer Darstellungen. Das Hauptmerkmal der 3D-Computergrafik besteht darin, dass räumliche Objekte als solche beschrieben bzw. modelliert und in räumlichen Koordinaten bzw. Gleichungen in räumlichen Koordinaten repräsentiert werden. Erst am Ende des Erstellungsprozesses einer dreidimensionalen computergrafischen Darstellung erfolgt die Abbildung (Projektion) der räumlichen Objekte in eine Ebene.

### 2.2.1 Ein einführendes Beispiel

Der 3D-Computergrafik liegt die Vorstellung einer virtuellen Kamera zugrunde, durch die ein Ausschnitt des dreidimensionalen Raumes „fotografiert“ wird. Das in Abb. 2.16 dargestellte Bild wurde mit der skriptgesteuerten 3D-Grafiksoftware POV-Ray durch die Eingabe folgender Szenenbeschreibung generiert:<sup>24</sup>

```
background {color rgb <1,1,1>}
camera { location <0,8,-20>
          angle 12
          look_at <0,-0.5,0> }
light_source { <-10,30,-10>
               color rgb <1,1,1>}
sphere { <-0.3,-0.5,1.5>, 0.5
         texture { pigment {color rgb <0,0,1>} } }
cone { <1.2,-1,0.5>, 0.5, <1.2,0.8,0.5>, 0
       texture { pigment {color rgb <1,0,0>} } }
torus {0.9, 0.15
       texture { pigment {color rgb <0,1,0>} }
       translate <-0.6,-0.85,-0.6> }
cylinder { <0,-1.1,0>, <0,-1,0>, 2
          texture { pigment {color rgb <0.5,0.5,0.5>}
                  finish {reflection 0.2} } }
```

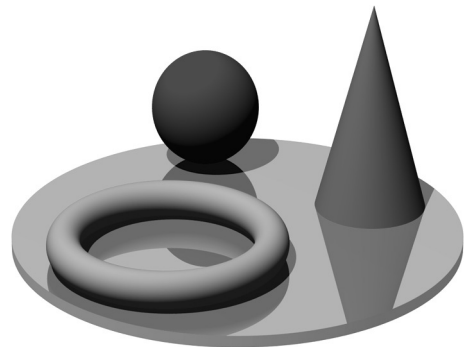


Abbildung 2.16: Beispiel einer dreidimensionalen Computergrafik

Die von der Software berechnete farbige Version dieses Bildes befindet sich auf S. 350 in Anhang B.

Alle Koordinatenangaben beziehen sich auf ein linkshändiges Koordinatensystem; die  $y$ -Achse wird als vertikale Achse abgebildet (siehe Abschnitt 2.3.1). In der Computergrafik ist diese Wahl des Koordinatensystems recht gebräuchlich.

<sup>24</sup>In dem Ordner 2-2 (siehe Anhang D, S. 365) befindet sich die Datei **Einf-Bsp.pov**, die den oben abgedruckten Quelltext enthält und aus der POV-Ray die farbige Version von Abb. 2.16 erzeugt.

Das auf S. 80 dargestellte Beispiel enthält alle Elemente, die für die Erzeugung einer dreidimensionalen computergrafischen Darstellung notwendig sind:

- Die Anweisung **background** legt die Farbe des Hintergrundes fest.<sup>25</sup>
- Hinter **camera** werden die Koordinaten einer Kamera (**location**), ihr Öffnungswinkel (**angle**) sowie die Koordinaten des Punktes, auf den diese Kamera gerichtet ist (**look\_at**), angegeben. Die Kamera „fotografiert“ die Szene, letztendlich erfolgt dadurch eine Zentralprojektion (siehe Abschnitt 2.4.3).
- Die Anweisung (**light\_source**) beschreibt eine Lichtquelle durch ihre Position im Raum und die Farbe des von ihr abgegebenen Lichtes. Da sich die Bildberechnung vieler 3D-Grafikprogramme an Gesetze der Optik anlehnt, müssen Körper beleuchtet werden, um sichtbar zu sein; es ist also mindestens eine Lichtquelle erforderlich. Einfache Lichtquellen werden als punktförmig angenommen.
- Die mit **sphere**, **cone**, **torus** und **cylinder** beginnenden Zeilen erzeugen geometrische Körper mit den dahinter angegebenen Abmessungen und positionieren sie (mit Ausnahme des Torus) durch Koordinaten bestimmender Punkte im Raum. Eine Kugel wird z. B. durch die Koordinaten ihres Mittelpunktes und den Radius beschrieben. Näher wird auf die Beschreibung geometrischer Körper durch Koordinaten und Abmessungen in Abschnitt 2.3.3 eingegangen. Neben einfachen Körpern kommen in der Computergrafik geometrische Objekte zur Anwendung, die Beschreibungen durch Gleichungen, Annäherungen von Punktmengen durch Interpolationspolynome oder konstruktive Methoden erfordern (siehe 2.3.5–2.3.9).
- Sichtbaren Körpern sind in der 3D-Computergrafik Oberflächeneigenschaften, allgemein als *Texturen* bezeichnet, zuzuweisen. In POV-Ray geschieht dies durch den Anweisungsblock **texture**, der innerhalb der Beschreibung sichtbarer Objekte (vor deren schließender Klammer) angeordnet ist. Im einfachsten Falle enthalten Texturdefinitionen Festlegungen von Farben, die in dem auf S. 80 dargestellten Beispiel durch RGB-Vektoren beschrieben sind. Oberflächenerscheinungen werden jedoch durch eine Reihe weiterer Merkmale gekennzeichnet, wie z. B. Rauheit und Reflexionsvermögen. So weist die Festlegung **reflection 0.2** dem Zylinder eine spiegelnde Oberfläche zu. Die Beschreibung optischer Eigenschaften von Oberflächen und die dem zugrunde liegenden mathematischen Modelle sind Gegenstände von Abschnitt 2.5.2.
- Nicht alle Objekte können bereits bei ihrer Erzeugung an der beabsichtigten Stelle positioniert oder auf die gewünschte Größe skaliert werden. So erzeugt POV-Ray einen Torus generell durch Rotation eines Kreises um die *y*-Achse und mit dem Zentrum im Koordinatenursprung. Durch die Anweisung **translate** wird der Torus an die vorgesehene Stelle verschoben. Oft sind auch Drehungen, Skalierungen und andere Transformationen auf Objekte anzuwenden (siehe 2.4.1).

---

<sup>25</sup>Die Angabe **color rgb <1,1,1>** beschreibt die Farbe Weiß im RGB-Modell (siehe S. 76f.). Auf die **background**-Anweisung könnte auch verzichtet werden, allerdings generieren die meisten Grafikprogramme dann einen schwarzen Hintergrund.

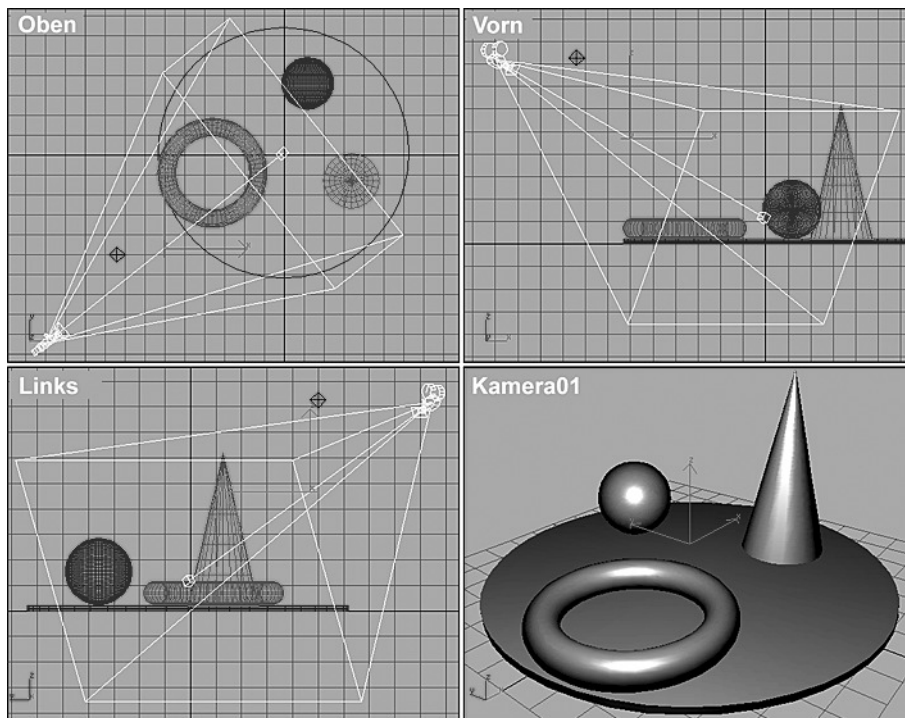


Abbildung 2.17:  
Einfache Szene in  
der Grafiksoftware  
3ds max

Unter einer *Szene* wird die Gesamtheit aller Elemente einer computergrafischen Darstellung aufgefasst: Kamera, Lichtquellen, sichtbare Objekte mit ihren geometrischen und optischen Eigenschaften sowie globale Gegebenheiten (Hintergrundfarbe, Nebel usw.).

Obwohl in den meisten 3D-Grafikprogrammen Szenen nicht durch textbasierte Szenenbeschreibungen (Skripte) eingegeben, sondern hauptsächlich mithilfe der Maus erzeugt und positioniert werden (siehe Abb. 2.17), sind dort dieselben Elemente zu erstellen.<sup>26</sup> Die auf S. 80 angegebene Szenenbeschreibung informiert auch über die in mausgesteuerter Software notwendigen Arbeitsschritte. So kann die Skriptzeile

```
camera { location <0,8,-20>   angle 12   look_at <0,-0.5,0> }
```

als Handlungsanweisung interpretiert werden:

*Erstelle eine Kamera im Punkt mit den Koordinaten (0;8;-20) mit dem Öffnungswinkel 12° und richte diese Kamera auf den Punkt mit den Koordinaten (0;-0,5;0) aus.*

Die anderen Bestandteile des Skripts lassen sich in ähnlicher Weise als „Konstruktionsanweisungen“ lesen. Da Skriptzeilen kürzer als verbale Darstellungen sind und die Skriptsprache von POV-Ray recht gut verständlich ist, wird diese in den folgenden Abschnitten auch für die Beschreibung von Objekten und Vorgehensweisen verwendet, die sich nicht auf die Nutzung einer speziellen Software beziehen.

<sup>26</sup>In üblichen 3D-Softwarepaketen (siehe Abschnitt 2.7.3) werden geometrische Objekte, Kameras und Lichtquellen mithilfe grafischer Oberflächen erzeugt und angeordnet. Dabei kann in orthogonalen und perspektivischen Ansichtsfenstern gearbeitet werden; in Abb. 2.17 sind die Vorder- und eine Seitenansicht, die Draufsicht sowie eine Kameraansicht dargestellt.



### 2.2.2 Arbeitsschritte der 3D-Computergrafik

Wie anhand des in Abschnitt 2.2.1 betrachteten Beispiels deutlich wird, sind für die Erstellung von 3D-Computergrafiken aus Anwendersicht folgende Aspekte maßgeblich:

- *Geometrische Modellierung*: Repräsentation, Transformation und gegebenenfalls Animation<sup>27</sup> von Körpern oder Flächen des Raumes durch Koordinaten bzw. Gleichungen in räumlichen Koordinaten;
- *Festlegung von Oberflächeneigenschaften* (Farbe sowie bei fotorealistischen Darstellungen auch Struktur und Reflexionseigenschaften) der Objekte;
- *Beleuchtung der Szene* (Definition von Lichtquellen);
- *Projektion der Szene in eine Ebene*, von der ein Ausschnitt auf dem Bildschirm dargestellt wird. Vom Anwender sind dazu die Kamerakoordinaten festzulegen.

Im Folgenden wird ein Überblick über die Berechnungen und Transformationen gegeben, die Grafikprogramme ausführen, um auf Grundlage der Benutzereingaben Bilder dreidimensionaler Szenen zu generieren.<sup>28</sup> Damit erfolgt eine (grobe) Einordnung der in den folgenden Abschnitten beschriebenen Verfahren der Computergrafik.

- *Umwandlung geometrischer Objekte in eine für die Weiterverarbeitung geeignete Repräsentation*. Viele Grafikprogramme wandeln geometrische Körper und Flächen vor der weiteren Verarbeitung in Polygonnetze (meist Dreiecksnetze) um. Da zudem die Bearbeitung von Polygonnetzen eine wichtige Modellierungstechnik in der 3D-Computergrafik ist, wird darauf im Zusammenhang mit der Modellierung geometrischer Objekte in Abschnitt 2.3.9 eingegangen.
- *Durchführung geometrischer Transformationen* (Abschnitt 2.4). Objekte können unter Verwendung „eigener“ Objektkoordinatensysteme modelliert werden. Durch Koordinatentransformationen, welche die Objektkoordinaten in die globalen Koordinaten der Szene transformieren, erhalten Objekte die gewünschte Lage innerhalb der Szene.<sup>29</sup> Da aus dreidimensionalen Szenen zweidimensionale Bilder erzeugt werden sollen, sind *Parallel- oder Zentralprojektionen in eine Ebene* erforderlich. Vorher werden Szenen oftmals durch *Clipping* auf das Sichtvolumen der Kamera eingeschränkt. Eine weitere (nur noch zweidimensionale) Transformation überführt das Bild in Pixelkoordinaten, ordnet also diskrete Punkte des durch Projektion entstandenen zweidimensionalen Bildes den Pixeln einer Bilddatei zu.

---

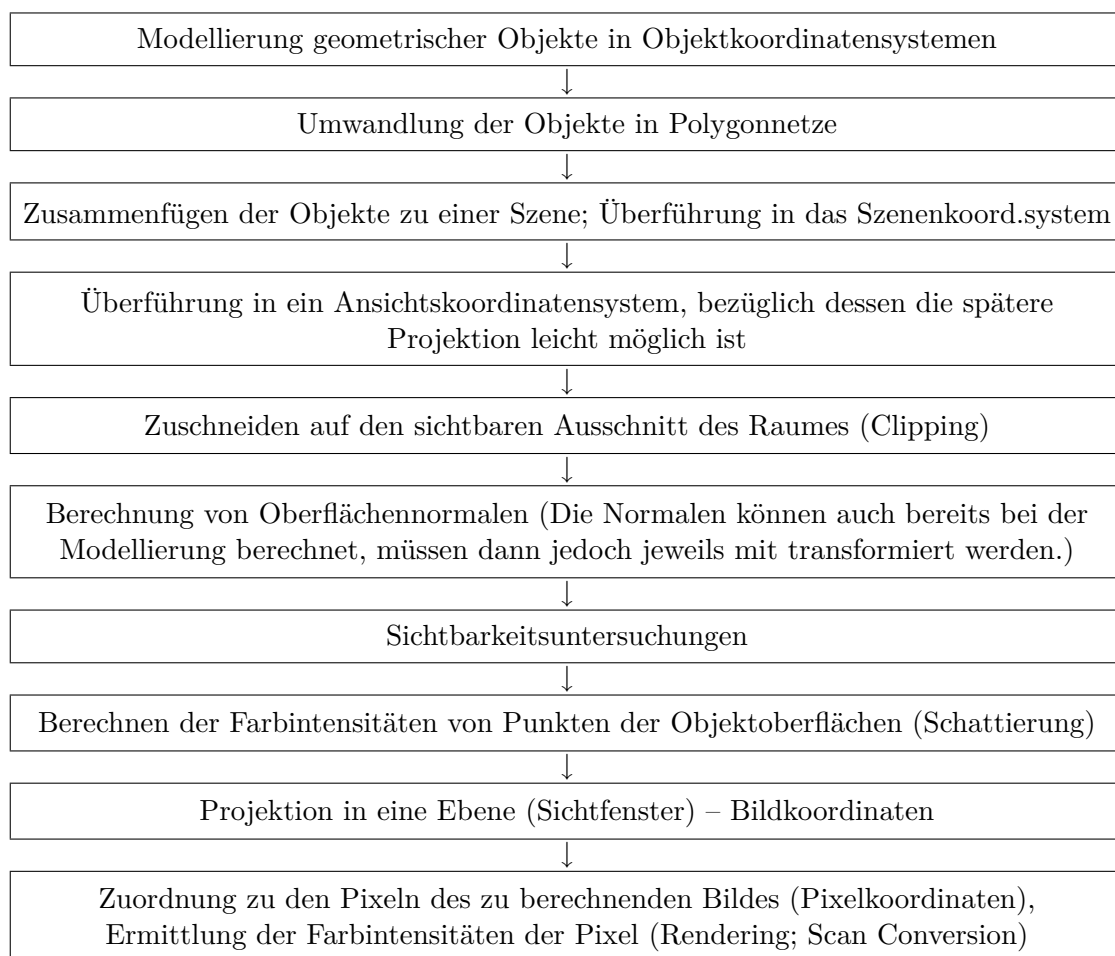
<sup>27</sup>Unter Animation (siehe Abschnitt 2.6) wird die zeitabhängige Veränderung von Szenen verstanden, wodurch Bildsequenzen bzw. Videos entstehen. Zeitabhängige Veränderungen können durch die Darstellung von Koordinaten, Größen und nicht geometrischen Werten (z. B. Farben) sowie der Kenngrößen geometrischer Transformationen als Funktionen eines Zeitparameters erreicht werden.

<sup>28</sup>Ein Verständnis dieser durch Software ausgeführten Arbeitsschritte ist nicht nur für Programmierer, sondern auch für Anwender von Bedeutung. Über eine Vielzahl von Parametern besteht bei allen leistungsfähigen 3D-Grafikprogrammen die Möglichkeit, Einfluss auf deren Arbeitsweise (Art der Projektion, Wahl der Schattierungsalgorithmen, Qualität der berechneten Bilder u. v. m.) zu nehmen.

<sup>29</sup>Diese Vorgehensweise lässt sich auch als Modellierung in einer ausgezeichneten Lage und anschließende geometrische Transformation interpretieren (wie im Falle des Torus in dem Beispiel auf S. 80).

- *Überprüfung der Sichtbarkeit* von Objekten bzw. Objektfacetten und *Entfernung verdeckter Kanten und Flächen*.
- *Berechnung der Helligkeits- bzw. Farbwerte der einzelnen Pixel des Bildes*. Dreidimensionale Grafikprogramme berechnen, oft unter Berücksichtigung optischer Gegebenheiten, möglichst „fotorealistische“ Abbildungen. Dieser Vorgang wird als *Rendering* oder auch als „Schattierung“ bezeichnet (Abschnitt 2.5).
- Für Sichtbarkeitstests und für die Untersuchung der Interaktion von Lichtstrahlen mit Körperoberflächen sind die *Normalen in Punkten bzw. Facetten von Oberflächen* zu bestimmen. Bei Flächen, die durch Polygon- bzw. speziell Dreiecksnetze angenähert werden, ist für eine realistische (glatte) Darstellung zudem eine *Interpolation von Normalen* innerhalb von Facetten notwendig (siehe 2.5.3).

Die Reihenfolge der genannten und in dem folgenden Schema zusammengefassten Schritte hängt von dem angewendeten Verfahren der Bildberechnung ab und kann variieren. Bei dem in Abschnitt 2.5.5 beschriebenen Raytracing-Verfahren werden beispielsweise die Projektion des Raumes in die Bildebene, Sichtbarkeitsuntersuchungen und die Schattierung miteinander kombiniert. Eine Umwandlung geometrischer Objekte in Polygonnetze ist bei einigen Raytracing-Programmen nicht erforderlich.



## 2.3 Modellierung geometrischer Objekte im Raum; geometrische Repräsentationen

### 2.3.1 Koordinatensysteme im Raum; kartesische und homogene Koordinaten

Punkte des dreidimensionalen Raumes lassen sich in der euklidischen Geometrie durch Koordinatentripel bezüglich eines kartesischen Koordinatensystems beschreiben. In welchen Richtungen die Achsen bei grafischen Darstellungen gezeichnet werden, ist aus mathematischer Sicht unerheblich,<sup>30</sup> wenn sich die Orientierung nicht unterscheidet;<sup>31</sup> Abb. 2.18 a) und b) unterscheiden sich nur durch die Position, aus der das Koordinatensystem betrachtet wird. In beiden Fällen handelt es sich um ein rechtshändiges (positiv orientiertes) Koordinatensystem;<sup>32</sup> die Anordnung der Achsen in Bezug auf die Reihenfolge  $x, y, z$  entspricht der mathematischen Standardorientierung.

In Computergrafik-Software werden oft linkshändige (negativ orientierte) Koordinatensysteme verwendet. Um Koordinaten bezüglich eines rechtshändigen Koordinatensystems in Koordinaten bezüglich eines linkshändigen Systems zu überführen, sind ungleichsinnige (die Orientierung ändernde) Koordinatentransformationen erforderlich. Die Transformation mit der Matrix

$$T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

überführt Koordinaten bezüglich des rechtshändigen Koordinatensystems nach Abb. 2.18 a) bzw. b) in solche bezüglich des linkshändigen Systems entsprechend Abb. 2.18 c); es gilt  $\det(T) = -1$ .

<sup>30</sup>Im deutschsprachigen Raum, insbesondere in Schulbüchern, dominieren Darstellungen, bei denen die  $z$ -Achse „nach oben“ zeigt, während in der englischsprachigen Literatur oft Darstellungen wie in Abb. 2.18 b) auftreten. In der Computergrafik wird die  $z$ -Achse als „Tiefenachse“ aufgefasst, was sich in Begriffsbildungen wie  $z$ -Buffer (Tiefenpuffer) und  $z$ -Clipping (Tiefenbeschneidung) niederschlägt, siehe die Abschnitte 2.4.4 und 2.4.5.

<sup>31</sup>Zwei Koordinatensysteme (bzw. die zugehörigen Basen) haben dieselbe Orientierung, wenn sie durch eine gleichsinnige Transformation (d. h. durch eine Transformation, die durch eine Matrix mit positiver Determinante beschrieben wird) auseinander hervorgehen.

<sup>32</sup>Die Bezeichnung „rechtshändig“ ergibt sich aus der „Rechte-Hand-Regel“: Werden Daumen, Zeigefinger und Mittelfinger der rechten Hand jeweils zueinander senkrecht gehalten, so repräsentiert der Daumen die  $x$ -Achse, der Zeigefinger die  $y$ -Achse und der Mittelfinger die  $z$ -Achse. Analog dazu gibt es eine „Linke-Hand-Regel“ für negativ orientierte Koordinatensysteme (wie in Abb. 2.18 c).

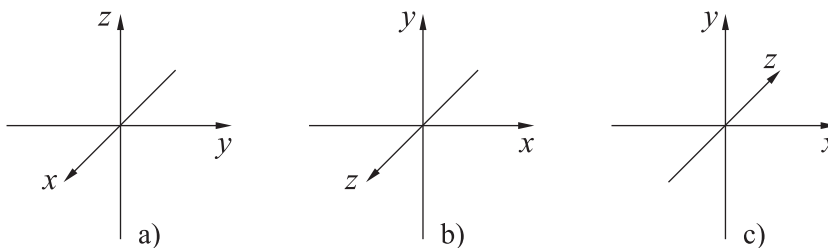


Abbildung 2.18: Rechtshändiges (a, b) und linkshändiges (c) Koordinatensystem

Für die praktische Verwendung eines Koordinatensystems zur Beschreibung und Modellierung geometrischer Objekte ist der Unterschied zwischen rechts- und linkshändigen Koordinatensystemen unbedeutend, so lange keine Transformationen zwischen verschiedenen Koordinatensystemen erforderlich sind. Der Übergang von ebenen zu räumlichen Koordinaten ist anhand des linkshändigen Koordinatensystems besonders intuitiv vollziehbar: Man betrachte ein ebenes Koordinatensystem in der geläufigen Darstellung und steche einen Pfeil in die Tafel- bzw. Papierebene.

### Homogene Koordinaten

In der Computergrafik ist neben der Nutzung kartesischer Koordinaten auch die Verwendung der aus der projektiven Geometrie bekannten homogenen Koordinaten gebräuchlich (vgl. u. a. [22], S. 63–66 und [197], S. 101–108). Dabei werden Punkte des Raumes durch Koordinatenquadrupel  $(x; y; z; w)$ , oft auch mit  $(x:y:z:w)$  bezeichnet, dargestellt. Die Zuordnung homogener Koordinaten zu Punkten ist bezüglich einer vorgegebenen Basis eindeutig bis auf Multiplikation mit einem von Null verschiedenen Skalar. Die Komponente  $w$  wird als *inverser Streckungsfaktor* bezeichnet und kann (bis auf  $w = 0$ ) frei gewählt werden, wodurch dann jedoch  $x$ ,  $y$  und  $z$  eindeutig festgelegt sind. Für  $w = 0$  werden uneigentliche, d. h. „unendlich ferne“ Punkte beschrieben.

Die euklidischen Koordinaten eigentlicher Punkte ergeben sich durch Multiplikation der zugehörigen homogenen Koordinaten mit  $w^{-1}$ . Den homogenen Koordinaten

$$(x; y; z; w)$$

entsprechen also die kartesischen Koordinaten

$$\left( \frac{x}{w}; \frac{y}{w}; \frac{z}{w} \right).$$

Standardmäßig wird der inverse Streckungsfaktor  $w = 1$  verwendet; in diesem Falle hat ein Punkt mit den euklidischen Koordinaten  $(x; y; z)$  die homogenen Koordinaten  $(x; y; z; 1)$ , mit denen in der Computergrafik häufig bei der Anwendung von Transformationen gearbeitet wird (siehe Abschnitt 2.4).

Homogene Koordinaten werden für interne Berechnungen in Computergrafik-Software genutzt, weil alle Transformationen des Raumes durch  $4 \times 4$ -Matrizen beschrieben werden können (die Darstellung von Verschiebungen durch  $3 \times 3$ -Matrizen bezüglich kartesischer Koordinaten ist nicht möglich) und Zentral- und Parallelprojektionen bezüglich homogener Koordinaten durch einheitliche Matrizen darstellbar sind. Des Weiteren bietet die Verwendung homogener Koordinaten in vielen Fällen numerische Vorteile, da durch die freie Wahl von  $w$  beliebige Punkte mit hinreichender Genauigkeit durch Ganzzahlen beschrieben werden können, deren Verarbeitung effizienter erfolgt als die von Gleitkommazahlen. Auf der Benutzerebene (für die Eingabe von Objekten in 3D-Grafiksoftware) werden jedoch kartesische Koordinaten verwendet; die Umsetzung in homogene Koordinaten erfolgt durch die verwendete Software ohne Interaktion mit dem Anwender. Die folgenden Betrachtungen zur Beschreibung dreidimensionaler Objekte werden daher auf der Grundlage kartesischer Koordinaten geführt.

### 2.3.2 Körper und Flächen

Die Modellierungstechniken in der 3D-Computergrafik lassen sich vom grundlegenden Ansatz her in die Modellierung von Körpern (in der Computergrafik oft auch als Volumenkörper bezeichnet) und die von Flächen unterteilen. Aus mathematischer Sicht sind Körper und Flächen klar unterschiedene Klassen von Objekten, z. B.:

*Kugeloberfläche* (mitunter auch *Sphäre* genannt): Menge aller Punkte des Raumes, die von einem vorgegebenen Punkt einen festen Abstand  $r$  (mit  $r > 0$ ) haben;

*Kugelkörper*: Menge aller Punkte des Raumes, deren Abstand von einem vorgegebenen Punkt kleiner oder gleich einem festen positiven Wert  $r$  ist.

Aus der Sicht der Computergrafik ist diese Unterscheidung oftmals nicht wesentlich, da Lichtstrahlen mit *Körperoberflächen* interagieren, eine Kugeloberfläche also z. B. nicht anders wahrgenommen wird als ein Kugelkörper. Eine Ausnahme bilden hierbei lediglich (teil-)transparente Körper, bei denen Lichtbrechung auftritt.

Ein weiteres Beispiel für die Beziehung zwischen Flächen und Körpern stellen die in der Computergrafik häufig auftretenden *Polygonnetze* (Flächen, die aus  $n$ -Ecken zusammengesetzt sind) sowie *Polyeder* dar, bei denen es sich um Körper handelt. Geschlossene Polygonnetze mit bestimmten Eigenschaften definieren Polyeder (siehe Abschnitt 2.3.9). Auf analoge Weise lassen sich auch durch Zusammensetzung gekrümmter Flächenstücke geometrische Körper erzeugen.

Sowohl die Modellierung mit geometrischen Grundkörpern sowie Vereinigungen, Durchschnitten und Differenzen von Körpern als auch die Beschreibung und Darstellung von Flächen sind für die Computergrafik bedeutsam.

### 2.3.3 Geometrische Grundkörper

Einfache geometrische Körper (Solid Primitives) werden durch wenige charakteristische Punkte und Abmessungen definiert (siehe obiges Beispiel der Kugeldefinition). In 3D-Grafikprogrammen können derartige Körper durch die Eingabe der Koordinaten bestimmender Punkte und der notwendigen Maße (bzw. durch Markieren der Punkte und „Ziehen“ der Abmessungen mit der Maus) beschrieben werden:<sup>33</sup>

Kugel: `sphere{<x,y,z>,r}` (Koordinaten des Mittelpunktes und Radius);

Gerader Kreiszylinder: `cylinder{<x1,y1,z1>,<x2,y2,z2>,r}`  
(Koordinaten der Mittelpunkte von Grund- und Deckfläche, Radius);

Gerader Kreiskegelstumpf: `cone{<x1,y1,z1>,r1,<x2,y2,z2>,r2}`  
(Koordinaten der Mittelpunkte sowie Radien von Grund- und Deckfläche);

Quader: `box{<x1,y1,z1>,<x2,y2,z2>}`  
(Koordinaten gegenüberliegender Eckpunkte bzgl. einer Raumdiagonalen; die Seitenflächen des Quaders werden parallel zu den Koordinatenebenen angeordnet);

<sup>33</sup>In den angegebenen Beschreibungen wird die Syntax der 3D-Grafiksoftware POV-Ray verwendet.

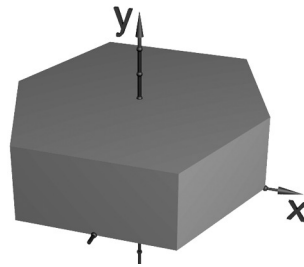
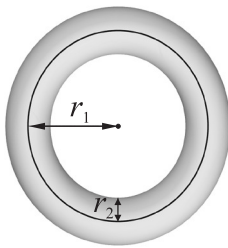


Abbildung 2.19: Torus

Datei: 2-3\Torus.pov

Abbildung 2.20: Prisma

Datei: 2-3\Prisma.pov

Torus: `torus{r1,r2}`

(Abstand des Ringmittelpunktes zum Mittelpunkt des Querschnittskreises, Radius des Querschnittskreises, siehe Abb. 2.19; der so beschriebene Torus wird in der  $x$ - $z$ -Ebene mit Mittelpunkt im Koordinatenursprung erzeugt);

Gerades Prisma: `prism{y1,y2,n,<x1,z1>,...,<xn,zn>}`

( $y$ -Koordinaten von Grund- und Deckfläche, welche parallel zur  $x$ - $z$ -Ebene erzeugt werden; Anzahl sowie  $x$ - und  $z$ -Koordinaten der Eckpunkte der Grundfläche).

Während für einige Körper durch die angegebenen Beschreibungen eine beliebige Anordnung im Raum möglich ist, werden andere (Tori, Quader, Prismen) in einer speziellen Lage erzeugt; auf diese müssen Transformationen (Verschiebungen und Drehungen, siehe Abschnitt 2.4) angewendet werden, um sie in die gewünschte Position zu bringen.

Viele 3D-Grafikprogramme nähern Körper mit gekrümmten Oberflächen durch Polygonnetze an.<sup>34</sup> Bei der Erzeugung derartiger Körper ist die Art und Feinheit der Approximation zu wählen. So können in der Software 3ds max Kugeln durch Vierecksnetze und verschiedene Arten von Dreiecksnetzen dargestellt werden; Abb. 2.21 zeigt als Beispiele zwei mit 3ds max erzeugte „Kugeln“, die durch geringe Anzahlen von Segmenten angenähert wurden. Durch eine genügend feine Unterteilung und eine bei der Bildberechnung erfolgende Kantenglättung (siehe Abschnitt 2.5.3) kann ein Aussehen erreicht werden, das sich von dem einer wirklichen Kugel nicht sichtbar unterscheidet.

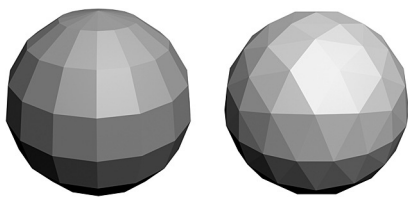


Abbildung 2.21: Erstellung von Kugeln in 3ds max

Die linke Abbildung zeigt eine mit 16 Segmenten erzeugte „Kugel“. Im rechten Bild wurde eine Kugel als Dreiecksnetz (in 3ds max als „Geosphäre“ bezeichnet) erstellt.

<sup>34</sup>Die Annäherung gekrümmter Oberflächen durch Polygonnetze erfolgt aus zwei Gründen. Die Berechnung von Schnittpunkten und Normalen erfordert bei ebenen Facetten nur das Lösen linearer Gleichungssysteme und deshalb recht wenig Rechenzeit. Außerdem ermöglicht es die Darstellung von Grundkörpern durch Polygonnetze, Formveränderungen durch Manipulationen von Eckpunkten vorzunehmen (Scheitelpunktmodellierung bzw. Vertex Modeling). Einfache Grundkörper bilden dabei den Ausgangspunkt für die Modellierung komplexerer Körper. Aus diesem Grunde gibt es Programme, die sowohl exakte Sphären, Zylinder usw. verarbeiten als auch die Erzeugung von angenäherten Polygonnetzen zur weiteren Bearbeitung ermöglichen.

### 2.3.4 Constructive Solid Geometry (Boolesche Operationen)

Eine leistungsfähige und intuitive Methode der Modellierung in der 3D-Computergrafik ist die Kombination von Körpern durch Boolesche Operationen (Vereinigung, Differenz und Durchschnitt). Bereits mithilfe der im vorangegangenen Abschnitt betrachteten Grundkörper lassen sich auf diese Weise vielfältige Formen erstellen – vor allem durch die Verknüpfung mehrerer Vereinigungen, Schnitte und Differenzen. Abb. 2.22 a) zeigt als Beispiel den Durchschnitt eines Prismas und einer Kugel, Abb. 2.22 b) die Differenz aus dem Durchschnitt zweier Zylinder und der Vereinigung zweier Zylinder:

```
difference{ intersection { cylinder {<-1,0,0>,<1,0,0>,1}
                                cylinder {<0,-1,0>,<0,1,0>,1}  }
merge { cylinder {<-1,0,0>,<1,0,0>,0.5}
        cylinder {<0,-1,0>,<0,1,0>,0.5}  }
} .
```

Als weiteres Beispiel ist in Abb. 2.22 c) das Ergebnis der Subtraktion einer Kugel vom Durchschnitt einer Kugel und eines Würfels dargestellt:

```
difference{ intersection { sphere {<0,0,0>,4}
                            box {<-3,-3,-3>,<3,3,3>}  }
sphere {<0,0,0>,3.5}
} .
```

In der Literatur wird die Modellierung durch Anwendung Boolescher Operationen auf Körper meist als *Constructive Solid Geometry* (CSG) bezeichnet. Voraussetzung für die Anwendung dieser Methode ist die Verwendung geometrischer Körper, deren Durchschnitt bzw. Differenz dreidimensional ist (auf die Vereinigung trifft dies ohnehin zu). Die Generierung von Schnittkurven zweier Flächen oder des Durchschnitts einer Fläche mit einem Körper ist mittels CSG nicht möglich.<sup>35</sup> In der Praxis können dennoch Schnittkurven und -flächen (wie z. B. Kegelschnitte) dargestellt werden, indem anstelle von Schnittebenen sehr dünne Quader zum Einsatz kommen.

<sup>35</sup>Die Dimension von Objekten muss bei CSG-Operationen erhalten bleiben. Es kommen daher *regularisierte Boolesche Operationen* zur Anwendung, die bei der Kombination von Körpern wieder Körper erzeugen. Dazu werden offene (bzw. teilweise offene) Punktmengen abgeschlossen und Randpunkte, die keinem inneren Punkt benachbart sind, entfernt (siehe [31], S. 195 und [79], S. 421). Beispielsweise ist der Durchschnitt zweier Würfel, die nur eine Seitenfläche gemeinsam haben, leer. In den meisten Fällen unterscheiden sich regularisierte Boolesche Operationen jedoch nicht von den bekannten Vereinigungen, Durchschnitten und Differenzen.

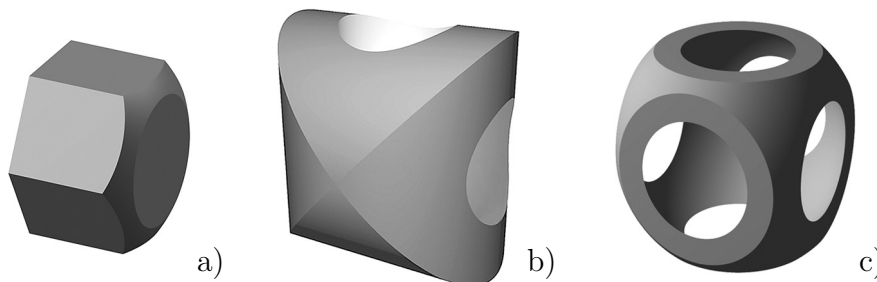


Abbildung 2.22:  
Boolesche Operationen

### 2.3.5 Beschreibung von Flächen durch Funktionsgleichungen, implizite Gleichungen und Parameterdarstellungen

#### Darstellung von Flächen durch Funktionsgleichungen

Flächen im Raum, die als Graphen von Funktionen zweier Variablen darstellbar sind, lassen sich durch Gleichungen der Form

$$z = f(x, y) \quad \text{bzw.} \quad y = f(x, z) \quad (2.1)$$

beschreiben. Die in Abb. 2.23 dargestellte Fläche wurde mittels der Gleichung

$$y = \sin x \cdot \sin z \quad (x, z \in [-10; 10])$$

in POV-Ray erzeugt.



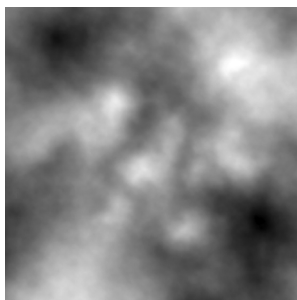
Abbildung 2.23: Graph einer Funktion zweier Variablen

Eine Farbversion dieser Abb. befindet sich in Anhang B auf S. 350.

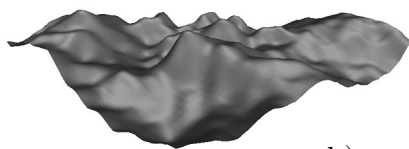
Eine erhebliche Einschränkung bei der Modellierung von Flächen durch Funktionsgleichungen besteht darin, dass jedem Paar von Argumenten nur ein Funktionswert zugeordnet werden und somit auf jeder zu der entsprechenden Koordinatenebene senkrechten Geraden maximal ein Punkt der Fläche liegen kann. Darstellungen von Flächen als Funktionsgraphen werden daher in der Computergrafik vor allem für die *Modellierung von Landschaften* genutzt. Dazu kommen jedoch i. Allg. keine Funktionen zum Einsatz, die sich in geschlossener Form durch Gleichungen beschreiben lassen. Neben fraktalen Funktionen finden vor allem Graustufenbilder Anwendung. Dem Helligkeitswert eines jeden Pixels eines Graustufenbildes wird dabei ein Höhenwert des zugehörigen Terrains zugeordnet; schwarze Pixel entsprechen meist den am niedrigsten gelegenen Punkten, weiße Pixel (mit dem normierten Helligkeitswert 1) den höchstmöglichen Erhebungen. Durch ein Graustufenbild (wie z. B. in Abb. 2.24 a) ist damit über einem Rechteck einer Grundebene ( $x$ - $z$ -Ebene) eine *Höhenfunktion*

$$y = f(x, z)$$

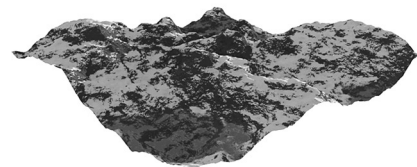
gegeben, durch die ein Terrain wie in Abb. 2.24 b) erzeugt werden kann. Neben der Höhe können durch die Helligkeitswerte der Bilddatei auch Farben bzw. Texturen des Terrainobjekts beeinflusst werden, wodurch sich z. B. niedrig liegende Punkte grün und hoch gelegene Punkte weiß darstellen lassen (siehe z. B. [119]). Allerdings sind recht komplizierte Funktionsterme notwendig, um zu realistisch wirkenden Landschaften zu



a)



b)



c)

Abbildung 2.24: Erzeugung eines Terrains durch Abbildung der Helligkeitswerte eines Graustufenbildes auf Höhenwerte

Entsprechende farbige Abb. befinden sich auf S. 350. Die Terrains wurden mithilfe der Dateien `Terrain.pov` und `Terrain.car` (siehe S. 365) in POV-Ray bzw. Carrara erzeugt.



gelangen; eine besondere Bedeutung haben Fraktale erlangt (siehe u. a. [124], S. 107–125 und [125], S. 234–257). Mithilfe der partiellen Ableitungen der Höhenfunktion können Punkten, in denen deren Anstieg sehr groß ist, graue Farbwerte zugeordnet werden, da steile Flächen in der Natur oft Felsen sind. Ein Beispiel für die Erzeugung komplexer Texturen auf der Grundlage der Höhenfunktion zeigt Abb. 2.24 c).

### Darstellung von Flächen durch implizite Gleichungen – Isosurfaces

Wie bereits erwähnt wurde, ist die Vielfalt der durch Funktionsgleichungen darstellbaren Flächen durch die notwendige Eindeutigkeit der Zuordnung  $(x, y) \rightarrow z$  beschränkt. Diese Beschränkung wird durch die Verwendung impliziter Gleichungen der Form

$$f(x, y, z) = 0 \quad (2.2)$$

umgangen. Ist dabei  $f$  eine lineare Funktion in den Variablen  $x$ ,  $y$  und  $z$ , so beschreibt die Gleichung eine Ebene, für eine quadratische Funktion eine Fläche zweiter Ordnung. Beliebige andere Klassen von Funktionen können ebenfalls im Funktionsterm von  $f$  auftreten. Als Beispiel zeigt Abb. 2.25 die mittels der Gleichung

$$\frac{\sin(10x)}{8} + \sqrt{z^2 + y^2} - 0,5 = 0$$

in POV-Ray und in dem Computeralgebrasystem MuPAD erzeugte Rotationsfläche.

a) POV-Ray:<sup>36</sup> `isosurface{ function{ sin(10*x)/8 + ((sqrt(z*z+y*y)) - 0.5)} contained_by {box{-2,2}}`

b) MuPAD: `plot::Implicit3d( sin(10*x)/8 + sqrt(z*z+y*y) - 0.5, x=-2..2, y=-0.7..0.7, z=-0.7..0.7, Mesh=[72,12,12] )`

### Darstellung von Flächen durch Parameterdarstellungen

Flächen im Raum können durch die Einführung zweier Parameter  $u$ ,  $v$  und die Darstellung der drei Raumkoordinaten als Funktionen

$$x = f_x(u, v), \quad y = f_y(u, v) \quad \text{sowie} \quad z = f_z(u, v) \quad (2.3)$$

dieser beiden Parameter beschrieben werden. Auch diese Beschreibung beinhaltet die

<sup>36</sup>Es sind nur die unbedingt notwendigen Eingaben aufgelistet; eine Anleitung mit weiteren Parametern und Optionen des `isosurface`-Objekts in POV-Ray enthält Anhang A (S. 337). Auf mathematische „Hintergründe“ der Flächendarstellung in CAS sowie in POV-Ray wird ab S. 93 eingegangen.

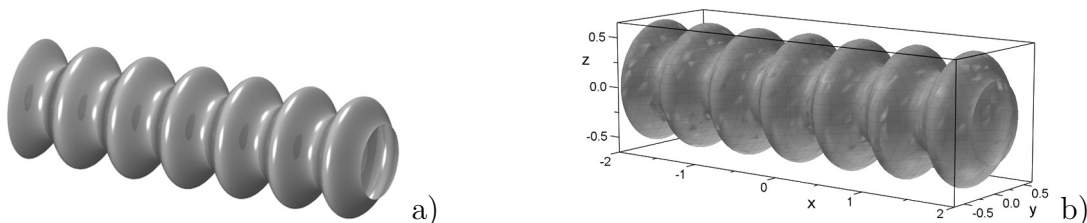


Abbildung 2.25: Darstellungen einer durch eine implizite Gleichung gegebenen Fläche. Entsprechende farbige Abb. enthält S. 350; die POV- bzw. MuPAD-Dateien, mit denen die Darstellungen generiert wurden, befinden sich in dem Ordner 2-3 (siehe Anhang D, S. 365).

Darstellung von Flächen durch Funktionsgleichungen als Spezialfall (z. B. mit  $u = x$ ,  $v = y$ ). Sind  $f_x$ ,  $f_y$  und  $f_z$  lineare Funktionen, so werden durch (2.3) Ebenen beschrieben (mit Ausnahme von Spezialfällen, in denen lediglich Geraden entstehen). Durch die Verwendung nichtlinearer Funktionen lässt sich eine große Vielfalt von Flächen modellieren.<sup>37</sup> Als Beispiel für eine durch Parametergleichungen beschriebene Fläche zeigt Abb. 2.26 mithilfe unterschiedlicher Softwarepakete erzeugte Darstellungen der durch

$$x = u \cdot v \cdot \sin(15v), \quad y = u \cdot v \cdot \cos(15v), \quad z = v \quad (u \in [0; 1], \quad v \in [-1; 1])$$

gegebenen konischen Spiralfäche. In den Grafikprogrammen POV-Ray und Carrara sowie in den CAS MuPAD und Mathematica sind dazu folgende Eingaben erforderlich:

- a) POV-Ray:<sup>38</sup>

```
parametric { function {u*v*sin(15*v)}  
              function {v}  
              function {u*v*cos(15*v)}  
              <0,-1>,<1,1> }
```
- b) Carrara:<sup>39</sup>

```
x = u*(2*v-1)*sin(15*(2*v-1));  
y = u*(2*v-1)*cos(15*(2*v-1));  
z = (2*v-1);
```
- c) MuPAD: 

```
plot::Surface( [ u*v*sin(15*v), u*v*cos(15*v), v ],  
               u=0..1, v=-1..1, UMesh=12, VMesh=192 )
```
- d) Mathematica: 

```
ParametricPlot3D[ { u*v*Sin[15*v], u*v*Cos[15*v], v },  
                  {u,0,1}, {v,-1,1}, PlotPoints {12,192} ]
```

<sup>37</sup> Parameterdarstellungen von Flächen sind Abbildungen aus  $\mathbb{R}^2$  ( $u$ - $v$ -Ebene) in  $\mathbb{R}^3$ , wobei zusammenhängenden Teilmengen der  $u$ - $v$ -Ebene Flächen des Raumes zugeordnet werden. Eigenschaften von Flächen (wie ihr Krümmungsverhalten) werden in der Differentialgeometrie anhand der Ableitungen der Funktionen  $f_x$ ,  $f_y$  und  $f_z$  beschrieben und untersucht (siehe z. B. [96]).

<sup>38</sup>Es sind nur die unverzichtbaren Eingaben aufgelistet; auf S. 338 in Anhang A werden weitere Parameter und Optionen des `parametric`-Objekts in POV-Ray erläutert.

<sup>39</sup>In der hauptsächlich mit der Maus zu bedienenden Software Carrara ist zunächst ein Formelobjekt („Formula Object“) zu erstellen; danach können in einem speziellen Fenster die Gleichungen eingegeben werden. Da Carrara für die Parameter generell den Definitionsbereich  $[0; 1]$  verwendet, ist in den Formeln  $2v-1$  anstelle von  $v$  einzusetzen, um die Fläche über  $u \in [0; 1]$ ,  $v \in [-1; 1]$  darzustellen.

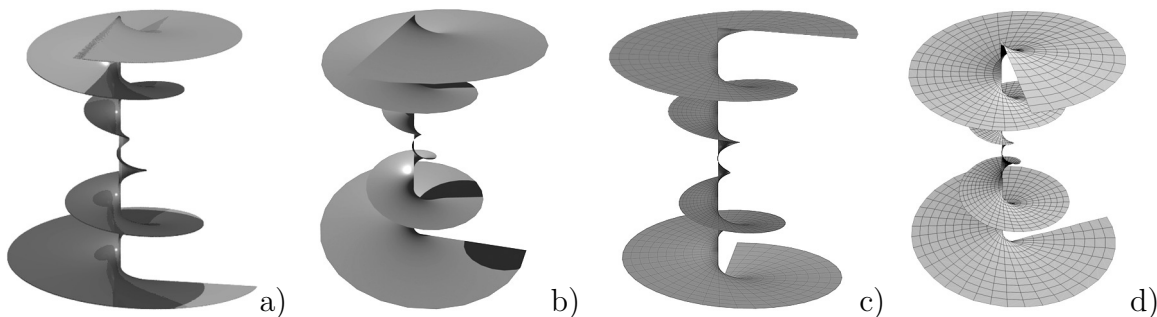


Abbildung 2.26: Darstellungen einer durch Parametergleichungen gegebenen Fläche. Farbige Versionen dieser Abbildungen enthält S. 351. Die Dateien, mit denen die Darstellungen in POV-Ray, Carrara, MuPAD sowie Mathematica generiert wurden, befinden sich in dem Ordner 2-3 (siehe Anhang D, S. 365).

### Software-interne Berechnungen und mathematischer Hintergrund der Darstellung von Flächen durch implizite Gleichungen und Parametergleichungen

Flächen, die durch Funktions-, implizite oder Parametergleichungen gegeben sind, werden in CAS und einigen Grafikprogrammen (z. B. Carrara) durch Polygonnetze angenähert. Dazu erfolgt eine Berechnung von Stützpunkten durch Unterteilung der Koordinaten- bzw. Parameterintervalle, über denen die Flächen definiert sind, in Teilintervalle. Die Feinheit dieser Unterteilung bestimmt die Genauigkeit der Darstellung und ist durch Anweisungen wie **Mesh** oder **Plotpoints** wählbar (siehe z. B. Abb. 2.26 und 2.27).

Um durch implizite Gleichungen der Form (2.2) gegebene Flächen darzustellen, ist es nötig, Stützpunkte durch das Lösen von Gleichungen zu bestimmen. Die Berechnungen benötigen dabei selbst für recht einfache nichtlineare Gleichungen i. Allg. wesentlich mehr Zeit als bei der Darstellung von Flächen, die durch Parametergleichungen gegeben sind. In letzterem Falle sind lediglich für eine festgelegte Zahl von Punkten in der  $u$ - $v$ -Ebene jeweils die Funktionswerte der Funktionen  $f_x$ ,  $f_y$  und  $f_z$  zu ermitteln. Um den Zeitaufwand für die Darstellung einer implizit gegebenen Fläche mit dem für eine Parameterdarstellung zu vergleichen, wurde eine Kugel einerseits durch die Gleichung

$$x^2 + y^2 + z^2 = 1 \quad (x, y, z \in [-1; 1])$$

und andererseits durch die Parameterdarstellung

$$f_x = \cos u \cos v, \quad f_y = \cos v \sin u, \quad f_z = \sin v \quad \left( u \in (-\pi; \pi], \quad v \in \left[-\frac{\pi}{2}; \frac{\pi}{2}\right] \right)$$

mithilfe des CAS MuPAD erzeugt. Bei einer vergleichbaren Anzahl von Stützpunkten<sup>40</sup> benötigte die Darstellung der parametrisch gegebenen Kugel etwa 1,5 Sekunden, für die durch eine implizite Gleichung gegebene Kugel wurden ca. 8 Sekunden benötigt.<sup>41</sup>

<sup>40</sup>Ein direkter Vergleich durch die Erzeugung identischer Stützpunkte ist nicht möglich, da die Stützpunkte bei impliziten Darstellungen durch Unterteilung der Koordinaten-, bei Parameterdarstellungen durch Unterteilung der Parameterintervalle bestimmt werden (siehe die Parameterlinien in Abb. 2.27). Um einen aussagekräftigen Vergleich zu erhalten, wurden die Unterteilungen so gewählt, dass für die parametrisch dargestellte Kugel mit **UMesh=60**, **VMesh=30** sogar eine etwas präzisere Annäherung erfolgte als für die implizit gegebene mit **Mesh=[12,12,12]**.

<sup>41</sup>Die Messungen wurden auf einem im Jahr 2004 aktuellen PC mit einem Pentium IV Prozessor, Taktfrequenz 3,2 GHz, durchgeführt. Die dazu verwendeten MuPAD-Dateien befinden sich in dem Ordner 2-3 (siehe Anhang D, S. 365).

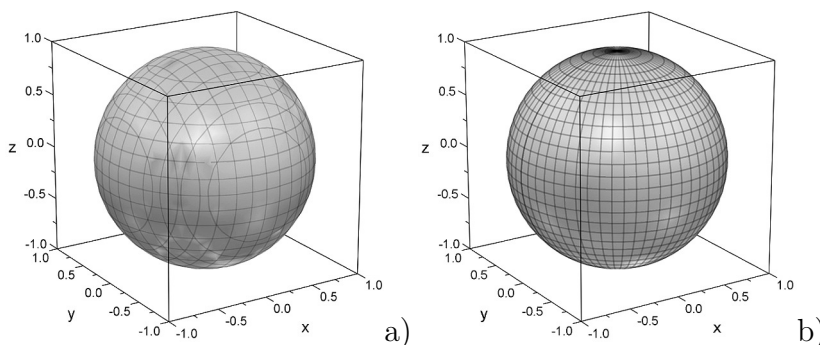


Abbildung 2.27: Darstellung einer Kugel in MuPAD durch eine implizite Gleichung (a) und eine Parameterdarstellung (b)

Es erscheint für das Beispiel der Kugel recht verwunderlich, dass für die implizite Darstellung wesentlich mehr Zeit als für die Parameterdarstellung benötigt wird. Schließlich kann die Kugelgleichung leicht nach jeder der Variablen  $x$ ,  $y$  und  $z$  umgestellt werden:

$$x = \pm\sqrt{1 - y^2 - z^2}, \quad y = \pm\sqrt{1 - x^2 - z^2}, \quad z = \pm\sqrt{1 - x^2 - y^2}.$$

Mithilfe dieser expliziten Gleichungen für jede Variable wäre eine sehr schnelle Berechnung der Stützpunkte möglich, die weniger Zeit in Anspruch nehmen würde als Stützpunktberechnungen anhand von Parametergleichungen. Allerdings handelt es sich bei der Kugel und ähnlich darstellbaren Flächen um Ausnahmen. Im allgemeinen Fall der Gleichung (2.2) ist eine Umformung in explizite Gleichungen für die einzelnen Variablen nicht möglich. CAS verwenden daher für die Berechnung der Eckpunkte der Fläche approximierenden Gitternetzes numerische Lösungsverfahren,<sup>42</sup> d. h. die Gleichung (2.2) muss für jedes Koordinatenpaar der festgelegten  $x$ - $y$ -,  $x$ - $z$ - und  $y$ - $z$ -Netze mittels numerischer Annäherung für die jeweils dritte Variable gelöst werden. Bereits ermittelte Lösungen dienen dabei als Ausgangswerte für die Berechnungen der jeweils nächsten Stützpunkte. Wie schon erwähnt wurde, benötigt das CAS MuPAD für dieses Verfahren bei einer Kugel etwa das Fünffache der Zeit, die für die Berechnung der Koordinaten einer vergleichbaren Anzahl von Stützpunkten durch Einsetzen der jeweiligen Parameterpaare in die Parametergleichungen der Kugel notwendig ist.

In Computergrafik-Software ist die Darstellung von Flächen anhand impliziter Gleichungen nur in Ausnahmefällen möglich.<sup>43</sup> Der höhere Zeitaufwand für implizite Darstellungen wirkt sich bei der Erzeugung fotorealistischer Grafiken und Videos wesentlich stärker aus als bei Darstellungen von Flächen in CAS. Das Programm POV-Ray wird jedoch im Gegensatz zur Mehrzahl der Grafikprogramme nicht mit dem Ziel einer optimalen Geschwindigkeit für Produktionsprozesse, sondern unter dem Hauptaspekt vielfältiger (auch experimenteller) grafischer Möglichkeiten und einer maximalen Darstellungsqualität entwickelt. Es verarbeitet sowohl implizite Gleichungen als auch Para-

<sup>42</sup>Obwohl CAS auch algebraische Umformungen vornehmen können, kommen für grafische Darstellungen hauptsächlich numerische Verfahren zum Einsatz. Algebraische Umformungen sind nicht für alle Gleichungen möglich und benötigen in den meisten Fällen mehr Zeit als numerische Näherungslösungen mit einer für grafische Darstellungen ausreichenden Genauigkeit. Eine Untersuchung auf Spezialfälle kann bei universell einsetzbaren Befehlen wie `plot::Implicit3d` nicht durchgeführt werden. Wenn jedoch Algorithmen für die Darstellung spezieller Flächen (wie z. B. Kugeln oder, etwas umfassender, Flächen 2. Ordnung) implementiert sind, so arbeiten diese i. Allg. wesentlich schneller als die universell verwendbaren Grafikbefehle, was anhand des hier betrachteten Beispiels auch nahe liegt.

<sup>43</sup>Darstellungen durch mathematische Beschreibungen gegebener Flächen gehören ohnehin nicht zu den Hauptanwendungsgebieten von CG-Software. Dennoch können einige Programme (wie Carrara sowie Maxon Cinema 4D, siehe Abschnitt 2.7.3) durch Parameterdarstellungen gegebene Flächen mittels Annäherung durch Gitternetze mit recht hoher Geschwindigkeit visualisieren. Aber auch Programme, die keine Objekte durch Eingabe mathematischer Terme erzeugen, arbeiten intern mit Parameterdarstellungen für Grundkörper, Bézier- und Splineobjekte (siehe 2.3.6 und 2.3.8) sowie für Gitternetze (2.3.9). Ein wichtiger Grund hierfür besteht darin, dass durch die Parametrisierung eine Abbildung aus der  $u$ - $v$ -Ebene in den Raum vorgenommen wird. Daraus ergibt sich die Möglichkeit, zweidimensionale Bilder auf Objektoberflächen zu projizieren und die Abbildung von Bildpunkten auf gewünschte Punkte der räumlichen Objekte präzise mithilfe von  $u$ - $v$ -Koordinaten zu beeinflussen (vgl. 2.4.6).

meterdarstellungen von Flächen, approximiert diese jedoch nicht durch Polygonnetze.

Dem in POV-Ray verwendeten Raytracing-Verfahren zur Bildberechnung (siehe Abschnitt 2.5.5) entsprechend, ermittelt dieses Programm für jeden berechneten Sehstrahl (vom Auge des Beobachters bzw. von der Kamera ausgehenden „Lichtstrahl“), der durch einen Bereich des Raumes verläuft, in dem die Fläche liegt, näherungsweise einen Schnittpunkt des Sehstrahls mit der Fläche.

Dazu ist ein Container-Objekt (Quader oder Kugel) anzugeben, innerhalb dessen die betrachtete Fläche liegt bzw. auf das die Darstellung der Fläche beschränkt werden soll. Die Strecke zwischen den Schnittpunkten des jeweils verfolgten Sehstrahls mit dem Container-Objekt wird dann sukzessive halbiert und für je 2 Unterteilungspunkte jeweils ermittelt, ob zwischen diesen beiden Punkten ein Schnittpunkt des Sehstrahls mit der Fläche liegen kann. Diese Intervallschachtelung wird abgebrochen, wenn die Intervalllänge einen vom Benutzer vorgegebenen Wert **accuracy** unterschreitet. Der gesamte Vorgang wird für alle Sehstrahlen wiederholt.

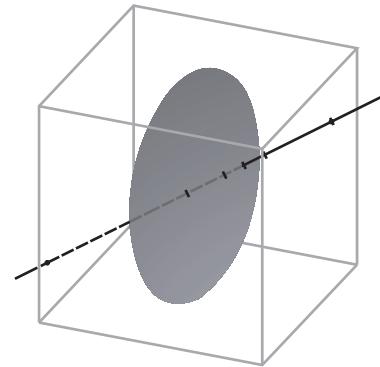


Abbildung 2.28: Fläche, Container-Objekt und Sehstrahl

Der entscheidende Schritt bei diesem Vorgehen zur Bestimmung von Punkten einer Fläche besteht darin, zu entscheiden, ob innerhalb einer Unterteilungsstrecke ein Punkt der Fläche liegen kann. Sind bei einer implizit durch eine Gleichung der Form (2.2), siehe S. 91, gegebenen Fläche zwei Unterteilungspunkte  $P_1(x_1; y_1; z_1)$ ,  $P_2(x_2; y_2; z_2)$  gegeben, so werden zunächst die Funktionswerte  $f(x_1, y_1, z_1)$  und  $f(x_2, y_2, z_2)$  berechnet. Ob auf der Strecke  $\overline{P_1P_2}$  ein Punkt der Fläche, also ein Punkt  $P(x; y; z)$  mit  $f(x, y, z) = 0$ , liegen kann, hängt von der „Änderungsrate“ (dem Gradienten) der Funktion  $f$  ab.<sup>44</sup>

<sup>44</sup>Als Gradient einer Funktion  $f(x, y, z)$  wird der Vektor der partiellen Ableitungen dieser Funktion nach  $x$ ,  $y$  und  $z$  bezeichnet:  $\text{grad} f = \nabla f = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial z} \end{pmatrix}$ . Von hoher Bedeutung sind Gradienten von Funk-

tionen u. a. in der Physik. Funktionen der Form  $f(x, y, z)$  (auch als Skalarfelder bezeichnet) ordnen Punkten des Raumes Skalare (z. B. Werte des Luftdrucks) zu.

Punktmengen mit gleichen Funktionswerten können als Niveaulinien, bei Funktionen zweier Variablen als Niveaulinien (wie z. B. auf Wetterkarten), dargestellt werden. Der Gradient einer Funktion ist eine Vektorfunktion, die Punkten Vektoren zuordnet, welche senkrecht auf den Niveaulinien bzw. -flächen durch die entsprechenden Punkte stehen. Je näher Niveaulinien einer Funktion in der Umgebung eines Punktes benachbart sind, desto schneller ändert sich der Funktionswert in der Nähe dieses Punktes und desto größer ist damit der Betrag des Gradienten. Abb. 2.29 zeigt Niveaulinien für die Funktionswerte 0; 0,2; 0,4; 0,6 und 0,8 der Funktion  $f$  mit  $f(x, y) = x^2 + y^2$ ; die Pfeile stellen Gradienten dieser Funktion in einigen Punkten dar.

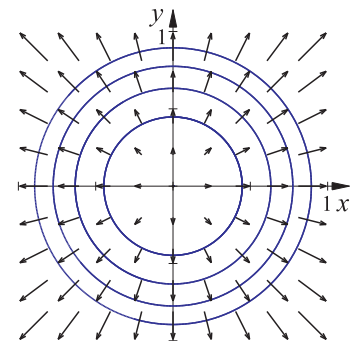


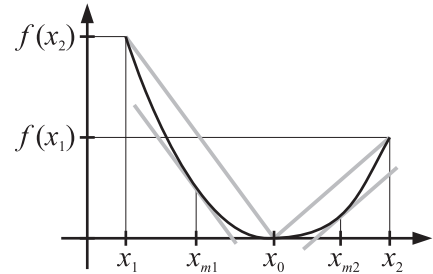
Abbildung 2.29: Niveaulinien und Gradientenvektoren

POV-Ray benötigt für die Darstellung einer durch eine implizite Gleichung<sup>45</sup> gegebenen Fläche die Angabe eines Wertes `max_gradient`, worunter eine obere Schranke für den Betrag  $\sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2 + \left(\frac{\partial f}{\partial z}\right)^2}$  des Gradienten verstanden wird.<sup>46</sup>

Die Überprüfung, ob auf der Strecke  $\overline{P_1P_2}$  ein Punkt der Fläche, also ein Punkt  $P(x; y; z)$  mit  $f(x, y, z) = 0$ , liegen kann, wird im Folgenden anhand des Spezialfalls beschrieben, dass die Strecke  $\overline{P_1P_2}$  auf der  $x$ -Achse liegt oder parallel zu dieser ist.<sup>47</sup>  $f$  kann dann als Funktion von  $x$  betrachtet werden, wobei  $f(x_1) = f(x_1, y_1, z_1)$  und  $f(x_2) = f(x_2, y_2, z_2)$  ist; o. B. d. A. sei  $x_1 < x_2$ . Da für den Fall, dass  $f(x_1)$  und  $f(x_2)$  unterschiedliche Vorzeichen haben, nach dem Zwischenwertsatz<sup>48</sup> bereits die Existenz einer Nullstelle (und somit eines Punktes der darzustellenden Fläche innerhalb der Strecke  $\overline{P_1P_2}$ ) gegeben ist, genügt es, die folgende Untersuchung für  $f(x_1) \cdot f(x_2) > 0$  durchzuführen.

Falls  $f$  eine Nullstelle  $x_0$  mit  $x_1 < x_0 < x_2$  hat, so existieren nach dem Mittelwertsatz der Differenzialrechnung Werte  $x_{m_1}$  und  $x_{m_2}$  mit

$$\frac{\partial f}{\partial x}(x_{m_1}) = \frac{f(x_0) - f(x_1)}{x_0 - x_1} = \frac{-f(x_1)}{x_0 - x_1} \quad \text{und} \\ \frac{\partial f}{\partial x}(x_{m_2}) = \frac{f(x_2) - f(x_0)}{x_2 - x_0} = \frac{f(x_2)}{x_2 - x_0}.$$



Ist  $g_{max} = \text{max\_gradient}$  eine obere Schranke für den Betrag des Gradienten, so können Absolutbeträge partieller Ableitungen nicht größer als  $g_{max}$  sein, es folgt also

$$\frac{|f(x_1)|}{x_0 - x_1} \leq g_{max} \quad \text{und} \quad \frac{|f(x_2)|}{x_2 - x_0} \leq g_{max} \quad \text{bzw.} \\ |f(x_1)| \leq g_{max} \cdot (x_0 - x_1) \quad \text{und} \quad |f(x_2)| \leq g_{max} \cdot (x_2 - x_0).$$

Da  $f(x_1)$  und  $f(x_2)$  gleiche Vorzeichen haben, ergibt sich durch Addition beider Ungleichungen

$$|f(x_1) + f(x_2)| \leq g_{max} \cdot (x_2 - x_1).$$

<sup>45</sup>Die Vorgehensweisen für Flächen, die durch Parameterdarstellungen gegeben sind, und Isosurfaces (durch implizite Gleichungen gegebene Flächen) ähneln sich. Allerdings erwartet POV-Ray bei Parameterflächen unter `max_gradient` die Angabe einer oberen Schranke für die Absolutbeträge der Funktionswerte aller partiellen Ableitungen  $\frac{\partial x}{\partial u}$ ,  $\frac{\partial x}{\partial v}$ ,  $\frac{\partial y}{\partial u}$ ,  $\frac{\partial y}{\partial v}$  sowie  $\frac{\partial z}{\partial u}$  und  $\frac{\partial z}{\partial v}$  über dem für  $(u, v)$  festgelegten Definitionsbereich. Im Folgenden wird auf das Verfahren bei Isosurfaces näher eingegangen.

<sup>46</sup>In vielen Fällen ist es nicht möglich, die partiellen Ableitungen der Funktion, die eine Fläche beschreibt, zu bestimmen. Die `isosurface`- und die `parametric`-Funktion von POV-Ray können jedoch auch durch schrittweises „Herantasten“ an optimale Werte verwendet werden. Außerdem existiert ein `evaluate`-Befehl, der Näherungswerte für `max_gradient` ermittelt (siehe [200]).

<sup>47</sup>Die Vorgehensweise im allgemeinen Fall ist analog. Dazu kann eine Variable  $l$  eingeführt werden, die Punkte auf der Geraden  $P_1P_2$  durch ihre (orientierten) Abstände zu einem vorgegebenen Punkt beschreibt.  $f(l)$  gibt dann die Funktionswerte von  $f$  für Punkte der Geraden  $P_1P_2$  an.

<sup>48</sup>Es wird im Folgenden die Stetigkeit und darüber hinaus die mindestens einmalige stetige Differenzierbarkeit der Funktion  $f$  nach allen Variablen vorausgesetzt. Flächen mit Singularitäten können ebenfalls dargestellt werden, jedoch treten an diesen Stellen Darstellungsfehler (meist „Löcher“) auf.

Ist diese Ungleichung für zwei Werte  $x_1, x_2$  nicht erfüllt, so kann  $f$  im Intervall  $[x_1, x_2]$  keine Nullstelle haben. Allgemein lässt sich (wie in Fußnote 47 vermerkt) herleiten:

*Falls für zwei Punkte  $P_1(x_1; y_1; z_1)$ ,  $P_2(x_2; y_2; z_2)$  die Funktionswerte  $f(x_1, y_1, z_1)$  und  $f(x_2, y_2, z_2)$  gleiche Vorzeichen haben und*

$$|f(x_1, y_1, z_1) + f(x_2, y_2, z_2)| > g_{\max} \cdot \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

*gilt (wobei  $g_{\max}$  eine obere Schranke für den Betrag des Gradienten von  $f$  ist), so enthält die Strecke  $\overline{P_1 P_2}$  keinen Punkt  $P_0(x_0; y_0; z_0)$  mit  $f(x_0, y_0, z_0) = 0$ , also keinen Punkt der durch  $f$  beschriebenen Fläche.*

In diesem Falle braucht die Strecke  $\overline{P_1 P_2}$  nicht weiter untersucht zu werden. Ansonsten wird der Mittelpunkt dieser Strecke berechnet und für jede der beiden Teilstrecken ermittelt, ob sie Punkte der Fläche enthalten können. Ist der für `max_gradient` =  $g_{\max}$  angegebene Wert zu groß, so werden überflüssige Teilstrecken untersucht. Dadurch verlängert sich die für die Darstellung einer Fläche benötigte Zeit. Für zu kleine Werte  $g_{\max}$  wird die Unterteilung zu früh abgebrochen, was zu Darstellungsfehlern führt.

Als Beispiel zeigt Abb. 2.30 in POV-Ray durch die Gleichung  $x^2 + y^2 + z^2 - 1 = 0$  dargestellte Kugeln. Der Gradient ist hierfür  $\text{grad} f = \begin{pmatrix} 2x \\ 2y \\ 2z \end{pmatrix}$ , sein Betrag  $2\sqrt{x^2 + y^2 + z^2}$ .

Offensichtlich ist dieser Wert für alle Punkte der Kugeloberfläche 2; insofern verwundert es nicht, dass das linke, mit der Einstellung `max_gradient 1.5` berechnete, Bild grobe Fehler aufweist. Allerdings ist auch das mittlere Bild (mit `max_gradient 2`) nicht fehlerfrei. Die Ursache hierfür besteht darin, dass die Unterteilung des Sehstrahls bei seinen Schnittpunkten mit dem Container-Objekt beginnt (siehe Abb. 2.28 auf S. 95). Dafür wurde in Abb. 2.30 ein durch die gegenüberliegenden Eckpunkte  $(-1, 1; -1, 1; -1, 1)$  und  $(1, 1; 1, 1; 1, 1)$  festgelegter Würfel verwendet, der Maximalwert für den Betrag des Gradienten ist also  $2\sqrt{1,1^2 + 1,1^2 + 1,1^2} \approx 3,81$ . Mit `max_gradient 2.2` erzeugt POV-Ray dennoch ein fehlerfreies Bild, da in dem in Abb. 2.30 dargestellten Beispiel durch die Punkte des Würfels, für die der Betrag des Gradienten der Funktion  $f$  besonders hoch ist, keine für die Abbildung der Kugel relevanten Sehstrahlen verlaufen.

```
isosurface {
    function { x*x + y*y + z*z - 1 }
    contained_by {box{ -1.1 , 1.1 }}
    max_gradient 1.5 // 2.0 2.2
    accuracy 0.0001
}
```

Wird der Wert `max_gradient` über das notwendige Maß hinaus erhöht, ergeben sich keine Veränderungen an dem erzeugten Bild, die Berechnungszeiten steigen jedoch erheblich.

<code>max_gradient</code>	2,2	3	5	10
Berechnungsdauer	5 s	7 s	10 s	15 s

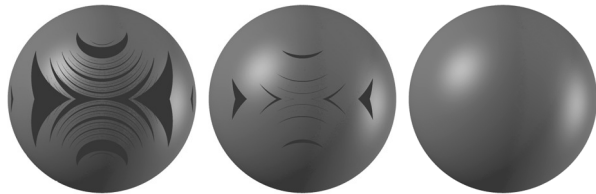


Abbildung 2.30: Einfluss des Wertes von `max_gradient` auf die Darstellungsqualität

Die Datei `Testsphere-MaxGradient.pov`, mit der die Abbildungen erzeugt und die Berechnungszeiten ermittelt wurden (Pentium IV Prozessor, 3,2 GHz), befindet sich in dem Ordner 2-3 (siehe S. 365).

### 2.3.6 Bézier- und Splinekurven

Kurven sind als eindimensionale Objekte in 3D-Darstellungen zwar nicht sichtbar, haben in der 3D-Computergrafik jedoch sowohl für die Konstruktion von Körpern bzw. Flächen durch Extrusion oder Rotation (siehe Abschnitt 2.3.7) als auch für die Beschreibung von Bewegungspfaden eine wichtige Bedeutung. Besonders häufig werden so genannte „Freiformkurven“ verwendet, die sich durch vorgegebene Punkte und Richtungen erzeugen lassen. Vor allem kommen dabei Bézierkurven, interpolierende kubische Splinekurven sowie B-Splines zur Anwendung.

#### Bézierkurven

Bézierkurven (und -flächen) entstanden aus den Bedürfnissen der Automobilindustrie nach computerunterstützten Entwurfsmöglichkeiten für Karosserien heraus und wurden von DE CASTELJAU (1959 bei Citroën, zunächst ohne Veröffentlichung) und BÉZIER (1961 bei Renault) unabhängig voneinander entwickelt. Sie erlangten in den folgenden Jahrzehnten Bedeutung in den unterschiedlichsten Bereichen. So sind beispielsweise die Fonts (Schriftzeichen), die in heutigen Computern zur Bildschirmdarstellung und zum Ausdruck von Dokumenten verwendet werden, Bézierkurven.

Von DE CASTELJAU wurde der im Folgenden beschriebene Algorithmus zur Erzeugung einer durch  $n + 1$  Kontrollpunkte beschriebenen Kurve angegeben.

Es seien  $n + 1$  Punkte  $P_0, \dots, P_n$  und eine reelle Zahl  $t$  mit  $0 \leq t \leq 1$  gegeben.

- Die Strecken  $\overline{P_k P_{k+1}}$  ( $0 \leq k \leq n-1$ ) werden im Verhältnis  $t$  geteilt, wobei  $n$  Teilungspunkte  $Q_0, \dots, Q_{n-1}$  entstehen (siehe Abb. 2.31).
- Die Strecken  $\overline{Q_k Q_{k+1}}$  ( $0 \leq k \leq n-2$ ) werden nun ebenfalls im Verhältnis  $t$  geteilt. Es entstehen  $n-1$  Teilungspunkte  $R_0, \dots, R_{n-2}$ , die Strecken  $\overline{R_k R_{k+1}}$  (mit  $0 \leq k \leq n-3$ ) bilden, welche wiederum im Verhältnis  $t$  geteilt werden usw..

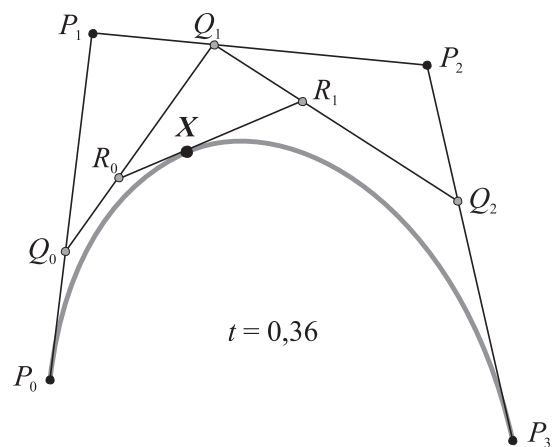


Abbildung 2.31: de-Casteljau-Algorithmus

Nach  $n-1$  Schritten bricht dieses Verfahren ab, es bleibt nur ein Punkt  $X$  übrig. Die Kurve, welche der Punkt  $X$  beschreibt, wenn  $t$  das Intervall  $[0; 1]$  durchläuft, wird als *Bézierkurve* bezeichnet.<sup>49</sup> Der erste und der letzte Kontrollpunkt  $P_0$  und  $P_n$  sind Punkte der Kurve, die Vektoren  $\overrightarrow{P_0 P_1}$  und  $\overrightarrow{P_n P_{n-1}}$  Tangentenvektoren an die Kurve

<sup>49</sup>Die Abhängigkeit der Kurve von den Kontrollpunkten und ihre Entstehung durch Veränderung des Teilverhältnisses  $t$  wird in der dynamischen Illustration *Bezierkurve-DeCasteljau-Alg.gsp* in dem Ordner 2-3 (siehe S. 365) deutlich, aus der auch Abb. 2.31 entstand. (Zum Öffnen dieser Datei ist die Software *The Geometer's Sketchpad* erforderlich.)



in diesen Punkten. Durch Verschieben der Punkte  $P_0$  und  $P_n$  sowie Veränderung der Tangentenvektoren durch „Ziehen“ an den „Anfassern“  $P_1$  und  $P_{n-1}$  lassen sich Bézierkurven interaktiv formen.

Die folgende Herleitung einer Parameterdarstellung für Bézierkurven beschränkt sich auf Bézierkurven mit 4 Kontrollpunkten ( $n = 3$ ), da sich diese – wie sich zeigen wird – durch Polynome 3. Grades beschreiben lassen und deshalb in der Computergrafik fast ausschließlich verwendet werden.<sup>50</sup> Die durch den de-Casteljau-Algorithmus erzeugten Punkte  $X$  einer Bézierkurve (siehe Abb. 2.31) lassen sich durch die Ortsvektoren

$$\vec{x} = \vec{r}_0 + t \cdot (\vec{r}_1 - \vec{r}_0) = (1-t) \cdot \vec{r}_0 + t \cdot \vec{r}_1$$

beschreiben. Analog gilt

$$\begin{aligned} \vec{r}_0 &= (1-t) \cdot \vec{q}_0 + t \cdot \vec{q}_1, & \vec{r}_1 &= (1-t) \cdot \vec{q}_1 + t \cdot \vec{q}_2, \\ \vec{q}_0 &= (1-t) \cdot \vec{p}_0 + t \cdot \vec{p}_1, & \vec{q}_1 &= (1-t) \cdot \vec{p}_1 + t \cdot \vec{p}_2 \quad \text{und} \quad \vec{q}_2 = (1-t) \cdot \vec{p}_2 + t \cdot \vec{p}_3. \end{aligned}$$

Durch jeweiliges Einsetzen der Gleichungen für  $\vec{q}_i$  in diejenigen für  $\vec{r}_i$  und schließlich der beiden resultierenden Gleichungen in die Darstellung von  $\vec{x}$  ergibt sich

$$\begin{aligned} \vec{x} &= (1-t)^3 \cdot \vec{p}_0 + 3t(1-t)^2 \cdot \vec{p}_1 + 3(1-t)t^2 \cdot \vec{p}_2 + t^3 \cdot \vec{p}_3 \\ &= \sum_{i=0}^3 \binom{3}{i} t^i (1-t)^{3-i} \cdot \vec{p}_i = \sum_{i=0}^3 B_{i,3}(t) \cdot \vec{p}_i. \end{aligned}$$

Die Koeffizienten  $B_{i,3}$  werden als *Bernsteinpolynome* (nach S. N. BERNSTEIN) bezeichnet. Für Bézierkurven mit  $n+1$  Kontrollpunkten lässt sich völlig analog

$$\vec{x} = \sum_{i=0}^n B_{i,n}(t) \cdot \vec{p}_i \quad \text{mit} \quad B_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i}. \quad (2.4)$$

herleiten. Zu den wichtigsten Eigenschaften von Bézierkurven zählt die bereits erwähnte Tatsache, dass die Verbindungsvektoren der jeweiligen beiden äußeren Kontrollpunkte Tangentenvektoren in den Endpunkten der Kurve sind. Als weitere Eigenschaft ist hervorzuheben, dass alle Punkte der Kurve innerhalb der konvexen Hülle des durch die Kontrollpunkte gebildeten Polygons liegen.<sup>51</sup>

Durch Zusammensetzen mehrerer der bisher betrachteten Bézierkurvenstücke mit kollinearen Tangentenvektoren in gemeinsamen Kontrollpunkten entstehen die in Vektorgrafiksoftware gebräuchlichen Bézierkurven mit beliebig vielen Kontrollpunkten und „Anfassern“ für Tangentenvektoren (siehe Abb. 2.4 auf Seite 70). Wird darüber hinaus gefordert, dass in den Übergangspunkten  $P_k$  zweier Kurvenstücke für die Tangentenvektoren  $\overrightarrow{P_k P_{k-1}}$  und  $\overrightarrow{P_k P_{k+1}}$  beider Kurvenstücke  $\overrightarrow{P_k P_{k-1}} = -\overrightarrow{P_k P_{k+1}}$  bzw.  $\overrightarrow{P_{k-1} P_k} = \overrightarrow{P_k P_{k+1}}$  gilt, so sind die zusammengesetzten Kurven in allen Punkten stetig differenzierbar; es handelt sich in diesem Falle also um Kurven der Differenzierbarkeitsklasse  $C_1$ .

<sup>50</sup>Die Gründe dafür, dass in der Computergrafik hauptsächlich kubische Kurven zum Einsatz kommen, bestehen darin, dass quadratische Funktionen keine genügend flexible Steuerung der Kurvenform ermöglichen und Polynome höherer Ordnung sowohl rechenintensiv als auch recht instabil sind (d. h., es können bei geringen Variationen der Kontrollpunkte starke Änderungen der Kurvenform auftreten).

<sup>51</sup>Auf Herleitungen dieser Eigenschaften wird hier verzichtet. Eine Einführung in Bézierkurven und ihre Eigenschaften enthält [53]; siehe außerdem [6], S. 348–366; [79], S. 383–389 und [257], S. 85–94.

## B-Splinekurven

Splinekurven wurden bereits vor den Bézierkurven genutzt; sie kamen u. a. im Schiffsbau zur Anwendung. Bei der Konstruktion eines Schiffsrumpfes werden lange, flexible Metallstreifen (Splines) an einzelnen Punkten fixiert.<sup>52</sup> Allgemein handelt es sich bei Splinekurven um stückweise zusammengesetzte Kurven, die in ihren Segmentgrenzen bestimmte Glattheitsbedingungen (Anforderungen an Stetigkeit bzw.  $k$ -fache stetige Differenzierbarkeit in den Übergangspunkten) erfüllen.<sup>53</sup> Als Funktionsterme, welche diese Kurven beschreiben, werden fast ausschließlich Polynome verwendet. Bézierkurven sind in diesem Sinne spezielle Splines. Daneben kommen in der Computergrafik noch andere bzw. verallgemeinerte Splinekurven zur Anwendung.

Bei *Basis-Splinekurven* (bzw. *B-Splinekurven* oder kurz *B-Splines*) besitzen Veränderungen von Kontrollpunkten nur lokalen Einfluss, ändern den Kurvenverlauf also nur in Bereichen um die betreffenden Punkte. Diese Eigenschaft ist für grafisches Konstruieren, bei dem oft Teilbereiche von Kurven zu modifizieren und anzupassen sind, von hoher Bedeutung. Eine B-Splinekurve vom Grad  $k-1$  wird durch  $n$  Kontrollpunkte (DE-BOOR-Punkte)  $P_1, \dots, P_n$  und einen Knotenvektor  $\vec{t} = \begin{pmatrix} t_1 \\ \vdots \\ t_{n+k} \end{pmatrix}$  (mit  $t_j \leq t_{j+1}$  für  $j = 1 \dots n+k-1$ ) festgelegt und durch folgende Parameterdarstellung beschrieben:

$$\vec{x}(t) = \sum_{i=1}^n N_{i,k}(t) \cdot \vec{p}_i \quad (t \in [t_1; t_{n+k})). \quad (2.5)$$

Dabei sind  $N_{i,k}(t)$  die Basisfunktionen der B-Splinekurve; sie werden rekursiv definiert:<sup>54</sup>

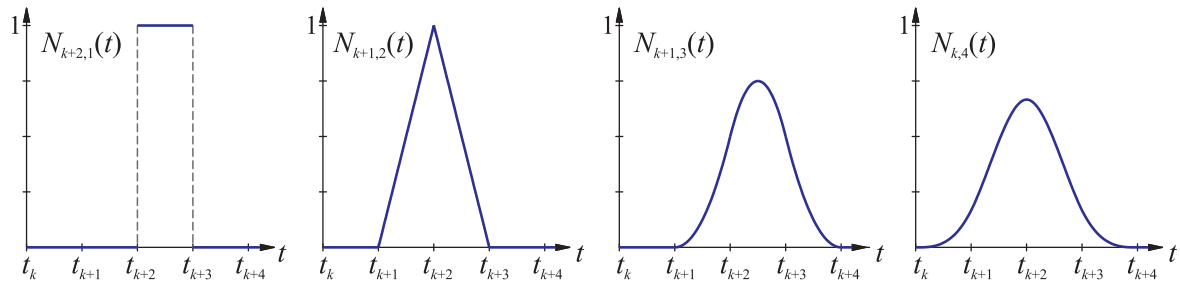
$$\begin{aligned} N_{i,1}(t) &= \begin{cases} 1 & \text{für } t_i \leq t < t_{i+1} \\ 0 & \text{sonst} \end{cases} \quad (1 \leq i \leq n+k-1), \\ N_{i,m}(t) &= \frac{(t - t_i)}{t_{i+m-1} - t_i} N_{i,m-1}(t) + \frac{(t_{i+m} - t)}{t_{i+m} - t_{i+1}} N_{i+1,m-1}(t) \quad \left( \begin{array}{l} 2 \leq m \leq k, \\ 1 \leq i \leq n+k-m \end{array} \right). \end{aligned}$$

B-Spline-Basisfunktionen vom Grad  $k-1$  (siehe Abb. 2.32) sind in höchstens  $k$  zusammenhängenden, von Komponenten des Knotenvektors  $\vec{t}$  begrenzten Intervallen von Null

<sup>52</sup>Steife Metallstreifen nehmen bei Fixierung durch Kontrollpunkte die Form an, bei der die geringste Biegeenergie auftritt. Dieser Fall tritt ein, wenn das Integral des Absolutbetrags der Krümmung über die gesamte Kurve, deren Form der Streifen annimmt, minimal ist. In [6], S. 341ff., wird gezeigt, dass interpolierende kubische Splinefunktionen diese Bedingung in guter Näherung erfüllen. Splinefunktionen sind nicht nur im Zusammenhang mit Konstruktionen und geometrischer Modellierung von Bedeutung, sondern vor allem in der Numerik, u. a. bei der Interpolation von Messwerten.

<sup>53</sup>In der Literatur werden Splinekurven mit recht unterschiedlichem Allgemeinheitsgrad definiert, siehe u. a. [6], S. 329–398; [11], S. 128–166; [31], S. 224–232; [79], S. 389–398 und [257], S. 94–113.

<sup>54</sup>Da an den Knotenvektor  $\vec{t} = (t_0; \dots; t_{n+k})^T$  nicht die Bedingung  $t_i \neq t_j$  für  $i \neq j$  gestellt wird, können die Terme  $t_{i+m-1} - t_i$  bzw.  $t_{i+m} - t_{i+1}$ , die bei  $N_{i,m}(t)$  im Nenner auftreten, den Wert Null annehmen. In diesen Fällen werden die gesamten betreffenden Summanden als Null aufgefasst. In der Definition wird auf diese Fallunterscheidung aus Gründen der Übersichtlichkeit verzichtet; bei der Umsetzung in einer Programmiersprache bzw. einem CAS ist sie unbedingt zu berücksichtigen.

Abbildung 2.32: B-Spline-Basisfunktionen 0., 1., 2. und 3. Grades ( $m = 1, 2, 3, 4$ )

verschieden.<sup>55</sup> Aus dieser Eigenschaft resultiert die Lokalität der B-Splinekurven, da die Änderung der Koordinaten eines Kontrollpunktes  $P_i$  die Parameterdarstellung (2.5) der Kurve nur über einem Intervall  $[t_i; t_{i+k}]$  verändert. Auf diesem Intervall werden die Basisfunktionen durch stückweise konstante, lineare, quadratische bzw. kubische Funktionsterme beschrieben (für  $k = 1, \dots, 4$ ).<sup>56</sup> Die Summe aller Basisfunktionen gleichen Grades ist eine konstante Funktion über dem gesamten Definitionsbereich von  $t$ :

$$\sum_{i=1}^n N_{i,k}(t) = 1 \quad (t \in [t_1; t_{n+k})). \quad (2.6)$$

B-Splines 3. Grades sind in allen einfachen Knoten ( $t_j$  mit  $t_{j-1} \neq t_j \neq t_{j+1}$ ) zweifach stetig differenzierbar (Kurven der Klasse  $C_2$ ), da die zweiten Ableitungen der kubischen Basisfunktionen stückweise lineare Funktionen und an einfachen Knoten stetig sind.

Den Einfluss des Knotenvektors  $\vec{t}$  auf den Verlauf einer kubischen B-Splinekurve zeigt Abb. 2.33. Die Kontrollpunkte beider Kurven sind identisch:  $P_1(-1; 1)$ ,  $P_2(-0,5; -1)$ ,  $P_3(0,4; -2)$ ,  $P_4(2; -0,5)$ ,  $P_5(3,6; -2)$ ,  $P_6(4,5; -1)$  und  $P_7(2; 2)$ . Die linke Kurve hat den Knotenvektor  $\vec{t} = (0; 0; 0; 0; 0,25; 0,5; 0,75; 1; 1; 1; 1)^T$ , für die rechte Kurve ist

<sup>55</sup>Auf die Darstellung der expliziten Funktionsterme der Basisfunktionen wird hier verzichtet; durch die hohe Zahl an Fallunterscheidungen für  $m \geq 3$  sind diese sehr umfangreich. Mithilfe der Dateien `B-Splines-MuPAD3.mnb` und `B-Splines-MuPAD4.mn` (für die MuPAD-Versionen 3 bzw. 4) in dem Ordner 2-3 (siehe S. 365) lassen sich die Funktionsterme für Basisfunktionen beliebigen Grades ausgeben.

<sup>56</sup>B-Splines höheren als 3. Grades werden in der Computergrafik aus den bereits in Fußnote 50 auf S. 99 genannten Gründen nur in sehr seltenen Ausnahmefällen genutzt.

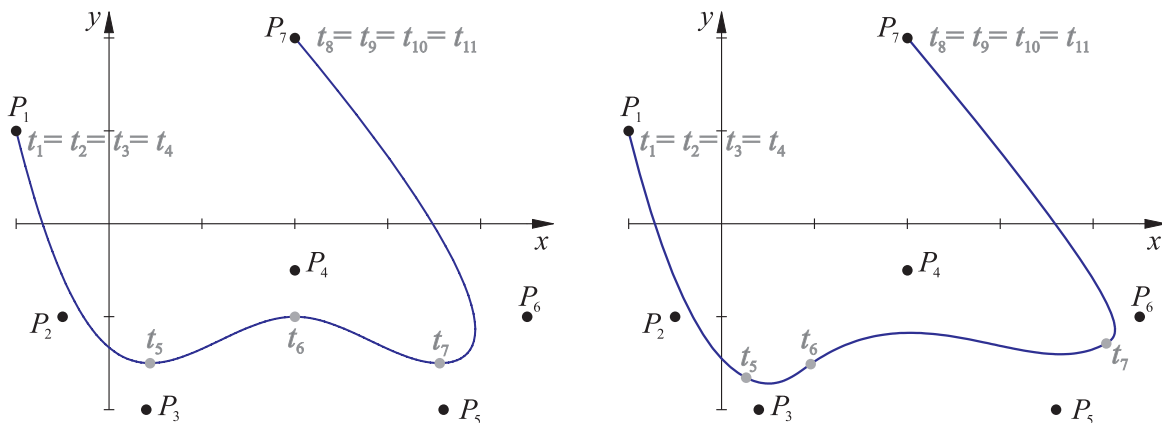


Abbildung 2.33: Uniforme (links) und nicht uniforme (rechts) B-Splinekurven

$\vec{t} = (0; 0; 0; 0; 0,1; 0,2; 0,85; 1; 1; 1; 1)^T$ . B-Splinekurven, bei denen die Parameterintervalle zwischen benachbarten Knoten (mit Ausnahme der identischen Randknoten) gleich lang sind, heißen *uniforme B-Splines*; bei unterschiedlich langen Parameterintervallen entstehen *nicht uniforme B-Splines* (siehe Abb. 2.33). Durch die Verwendung unterschiedlicher Knotenvektoren ergeben sich vielfältige Kurvenverläufe von B-Splinekurven (siehe u. a. [11], S. 156–163 und [257], S. 97–108).<sup>57</sup>

Die Möglichkeit, zusätzliche Kontroll- sowie Knotenpunkte einzufügen oder zu verschieben, ohne dabei den Kurvenverlauf global zu verändern, ermöglicht sehr flexible Kurvenanpassungen und ist die Hauptursache dafür, dass B-Splinekurven und -flächen (siehe Abschnitt 2.3.8) zu den wichtigsten Modellierungsmitteln in der professionellen 3D-Computergrafik gehören. Eine besonders große Bedeutung haben die *Nicht Uniformen Rationalen B-Splines* (Non-Uniform Rational B-Splines bzw. NURBS) erlangt. Diese stellen eine aus der Verwendung homogener Koordinaten (siehe S. 86) resultierende Verallgemeinerung der B-Splines dar. NURBS-Kurven können ebenfalls durch die Parameterdarstellung (2.5) auf S. 100 mit den rekursiv definierten B-Spline-Basispolynomen  $N_{i,k}(t)$  beschrieben werden. Die Ortsvektoren  $\vec{p}_i$  der Kontrollpunkte enthalten dabei wegen der Verwendung homogener Koordinaten jeweils eine zusätzliche Komponente  $w_i$ ; kartesischen Koordinatentripeln  $(x_i; y_i; z_i)$  entsprechen homogene Koordinatenquadrupel  $(w_i x_i; w_i y_i; w_i z_i; w_i)$ .<sup>58</sup> Für die  $w$ -Koordinaten von Punkten einer NURBS-Kurve gilt wegen (2.5)  $w(t) = \sum_{i=1}^n N_{i,k}(t) \cdot w_i$ . Durch Division der Parameterdarstellung (2.5) durch  $w(t)$  ergibt sich die folgende Parameterdarstellung einer NURBS-Kurve in kartesischen Koordinaten:

$$\vec{x}(t) = \frac{\sum_{i=1}^n N_{i,k}(t) \cdot w_i \cdot \vec{p}_i}{\sum_{j=1}^n N_{j,k}(t) \cdot w_j} \quad (t \in [t_1; t_{n+k})). \quad (2.7)$$

Die Parameterdarstellungen von NURBS sind für jede Komponente stückweise gebrochen rationale Funktionen.<sup>59</sup> Die aus der homogenen Darstellung hervorgegangenen Koeffizienten  $w_i$  wichten die Einflüsse von Kontrollpunkten auf den Verlauf der Kurve; sie werden daher als *Punktgewichte* bezeichnet.<sup>60</sup> Somit ermöglichen NURBS besonders vielfältige Möglichkeiten, den Kurvenverlauf zu beeinflussen: Anzahl und Positionen der Kontrollpunkte, Knoten, Punktgewichte. Daraus erklärt sich ihre stark gewachsene Bedeutung in den Bereichen der Computergrafik und des CAD.

<sup>57</sup>Beliebige ebene B-Splinekurven lassen sich durch Variation der Anzahl und der Koordinaten der Kontrollpunkte sowie des Knotenvektors in den auf S. 101 erwähnten MuPAD-Dateien generieren.

<sup>58</sup>Da im Allgemeinen eigentliche (euklidische) Kontrollpunkte betrachtet werden, sind alle  $w_i$  von Null verschieden. Damit kann durch Multiplikation jeweils aller Koordinaten der Kontrollpunkte mit geeigneten Faktoren erreicht werden, dass alle  $w_i$  positiv sind, wovon im Folgenden ausgegangen wird.

<sup>59</sup>Durch NURBS-Darstellungen lassen sich viele Kurven exakt beschreiben. So können z. B. Ellipsen, Parabeln und Hyperbeln als spezielle NURBS-Kurven 2. Grades aufgefasst werden (siehe [195], S. 109).

<sup>60</sup>Haben alle Punktgewichte  $w_i$  den Wert 1, so nimmt (2.7) wegen (2.6) die Gestalt (2.5) an. Die NURBS-Kurve ist in diesem speziellen Falle also eine gewöhnliche B-Splinekurve, was sich auch daraus erklärt, dass kartesische Koordinaten von Punkten aus deren homogenen Koordinaten durch Weglassen der  $w$ -Komponente entstehen, falls  $w = 1$  ist (siehe S. 86).

### Interpolierende Splinekurven

B-Splines (und auch die allgemeineren NURBS) verlaufen nur in Ausnahmefällen durch ihre Kontrollpunkte. Werden Kurven benötigt, welche die vorgegebenen Punkte enthalten, so kommen *interpolierende kubische Splinekurven* zum Einsatz. Für die Konstruktion einer derartigen Kurve sind  $n$  Stützpunkte  $P_1, \dots, P_n$  sowie zwei weitere Punkte  $P_0, P_{n+1}$ , die den Kurvenverlauf in den Endpunkten  $P_1$  und  $P_n$  bestimmen, anzugeben. Durch diese Punkte soll eine aus kubischen Kurvenstücken zusammengesetzte Kurve gelegt werden, die in jedem Punkt stetig differenzierbar (glatt) ist. Ein häufig (u. a. in POV-Ray und in Microsoft Word) verwendeter Ansatz zur Generierung von Kurven, die diesen Bedingungen genügen, stammt von CATMULL und ROM (siehe z. B. [11], S. 139). Bei CATMULL-ROM-Splines verlaufen die Tangenten in beliebigen Stützpunkten  $P_i$  parallel zu den Sekanten  $P_{i-1}P_{i+1}$  durch die jeweils benachbarten Stützpunkte, Abb. 2.34 zeigt dies anhand einiger Tangenten. Für die Tangentenvektoren  $\vec{p}_i'$  gilt somit

$$\vec{p}_i' = T \cdot (\vec{p}_{i+1} - \vec{p}_{i-1}) \quad (1 \leq i \leq n, T \in \mathbb{R}, T \geq 0), \quad (2.8)$$

wobei  $T$  (die Tension bzw. Spannung) den Einfluss der Tangenten auf den Kurvenverlauf angibt; oft wird für alle Stützpunkte  $T = \frac{1}{2}$  gesetzt. Jedes kubische Kurvenstück zwischen zwei Punkten  $P_i$  und  $P_{i+1}$  kann durch eine Parameterdarstellung der Form

$$\vec{p}(t) = \vec{a}_0 + t \cdot \vec{a}_1 + t^2 \cdot \vec{a}_2 + t^3 \cdot \vec{a}_3 \quad (2.9)$$

beschrieben werden, wobei im Sinne einer einfachen Bestimmung der Koeffizientenvektoren  $\vec{a}_0, \vec{a}_1, \vec{a}_2$  und  $\vec{a}_3$  der Parameter  $t$  auf dem Intervall  $[0; 1]$  betrachtet, also  $\vec{p}(0) = \vec{p}_i$  und  $\vec{p}(1) = \vec{p}_{i+1}$  gesetzt wird. Daraus und aus der Tangentenbedingung (2.8) ergeben sich durch Einsetzen in (2.9) folgende Bedingungen für die Koeffizientenvektoren:

$$\begin{aligned} \vec{p}_i &= \vec{p}(0) = \vec{a}_0 \\ \vec{p}_{i+1} &= \vec{p}(1) = \vec{a}_0 + \vec{a}_1 + \vec{a}_2 + \vec{a}_3 \\ T \cdot \vec{p}_{i+1} - T \cdot \vec{p}_{i-1} &= \vec{p}_i' = \vec{p}'(0) = \vec{a}_1 \\ T \cdot \vec{p}_{i+2} - T \cdot \vec{p}_i &= \vec{p}_{i+1}' = \vec{p}'(1) = \vec{a}_1 + 2\vec{a}_2 + 3\vec{a}_3. \end{aligned}$$

Durch Lösen dieses Gleichungssystems ergibt sich

$$\begin{pmatrix} \vec{a}_0 \\ \vec{a}_1 \\ \vec{a}_2 \\ \vec{a}_3 \end{pmatrix} = A \cdot \begin{pmatrix} \vec{p}_{i-1} \\ \vec{p}_i \\ \vec{p}_{i+1} \\ \vec{p}_{i+2} \end{pmatrix} \quad \text{mit} \quad A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -T & 0 & T & 0 \\ 2T & T-3 & 3-2T & -T \\ -T & 2-T & T-2 & T \end{pmatrix}.$$

Die Parameterdarstellung (2.9) erhält damit für beliebige Kurvenstücke  $\widehat{P_i P_{i+1}}$  einer Catmull-Rom-Splinekurve die Gestalt

$$\vec{p}(t) = (1; t; t^2; t^3) \cdot A \cdot \begin{pmatrix} \vec{p}_{i-1} \\ \vec{p}_i \\ \vec{p}_{i+1} \\ \vec{p}_{i+2} \end{pmatrix}.$$

Der Ordner 2-3 enthält MuPAD-Dateien, die es ermöglichen, Catmull-Rom-Splinekurven mit beliebigen Stützpunkten zu erzeugen und den Einfluss der Tension  $T$  zu untersuchen.

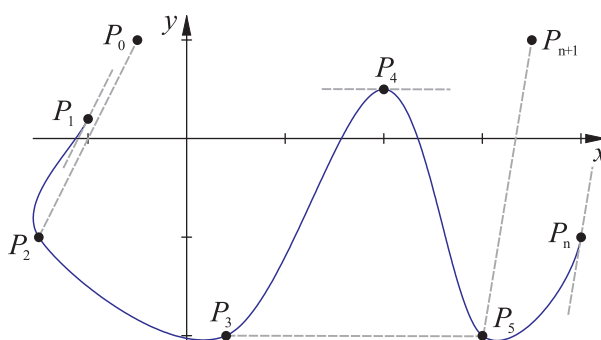


Abbildung 2.34: Catmull-Rom-Splinekurve

### 2.3.7 Konstruktion von Flächen und Körpern aus Kurven

Eine häufig genutzte Möglichkeit der Modellierung räumlicher Objekte besteht darin, Flächen durch *Extrusion* von (meist geschlossenen) Kurven zu erzeugen; hierbei wird eine Kurve an einer anderen Kurve (der Extrusionskurve) entlang „gezogen“; Abb. 2.35 zeigt als Beispiel die Extrusion eines Kreises entlang einer Bézierkurve.

Kommt als Extrusionskurve eine Gerade (bzw. Strecke) zur Anwendung, so entsteht ein verallgemeinerter Zylinder (in POV-Ray allerdings als „Prisma“ bezeichnet). Abb. 2.36 zeigt einen auf diese Weise erzeugten Extrusionskörper, dessen Grundfläche von einer interpolierenden kubischen Splinekurve (siehe S. 103) mit den Stützpunkten  $P_0, \dots, P_{16}$  begrenzt wird. Die zur Illustration eingefügte „Kurve“ entstand analog zu Abb. 2.35, wofür in POV-Ray das `sphere_sweep`-Objekt genutzt wurde (einhüllende Fläche einer sehr hohen Anzahl kleiner Kugeln entlang einer Kurve).

Häufig werden Extrusionen von Bézierkurven zur Erzeugung dreidimensionaler Schriftzüge verwendet. (Wie auf Seite 98 ausgeführt wurde, handelt es sich bei Schriften um Bézierkurven.)

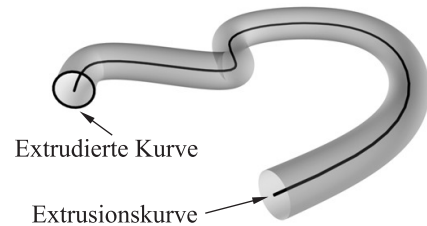


Abbildung 2.35: Extrusion

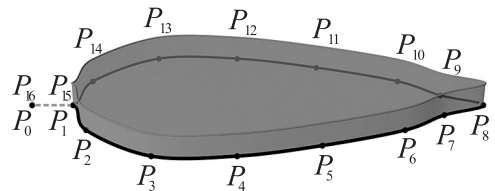


Abbildung 2.36: Zylinder über einer kubischen Splinekurve

Weitere Beispiele für die Konstruktion räumlicher Objekte aus Kurven sind *Rotationsflächen* bzw. *-körper*,<sup>61</sup> die durch Extrusion von Kurven entlang von Kreisen entstehen. Rotationsflächen sind somit spezielle Extrusionsflächen; gleichzeitig können sie als Flächen aufgefasst werden, die durch Rotation einer Kurve um eine Achse entstehen. Die Achse einer Rotationsfläche verläuft durch den Mittelpunkt des Extrusionskreises und ist senkrecht zu der Ebene durch diesen Kreis.

Analytisch lässt sich eine Fläche, die durch Drehung einer Kurve mit der Parameterdarstellung

$$\vec{p}(v) = \begin{pmatrix} x_K(v) \\ y_K(v) \end{pmatrix}; \quad v \in [v_1; v_2]$$

um die  $y$ -Achse entsteht, durch die Parameterdarstellung

$$\vec{r}(u, v) = \begin{pmatrix} x(u, v) \\ y(u, v) \\ z(u, v) \end{pmatrix} = \begin{pmatrix} x_K(v) \cdot \cos(u) \\ y_K(v) \\ x_K(v) \cdot \sin(u) \end{pmatrix}$$

(mit  $u \in [0; 2\pi]$ ,  $v \in [v_1; v_2]$ ) beschreiben.

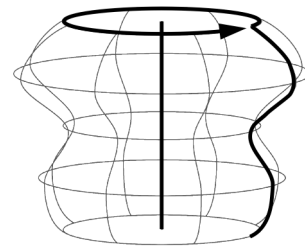
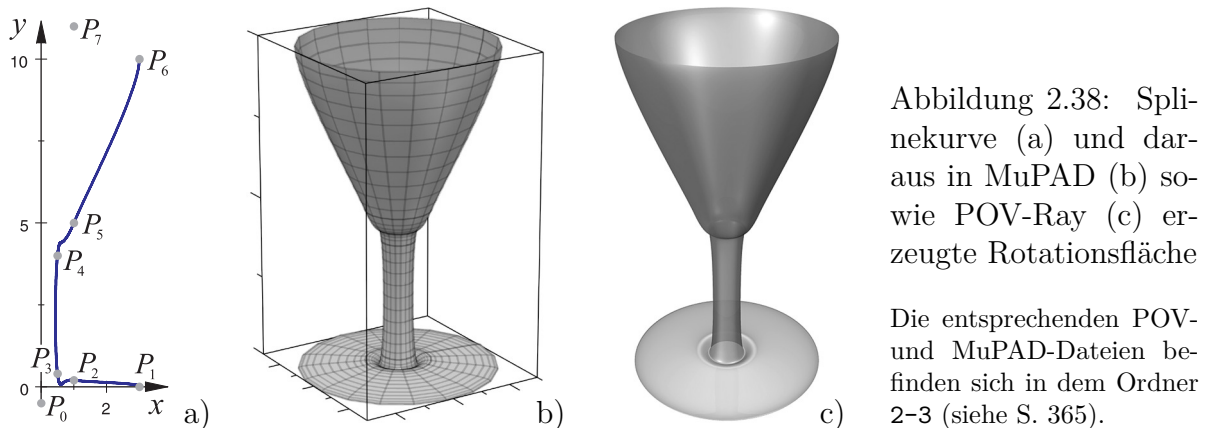


Abbildung 2.37: Rotationsfläche

<sup>61</sup>Flächen, die durch Rotation geschlossener Kurven entstehen, begrenzen Körper. Bei Rotation offener Kurven entstehen Körper, falls die Kurvenenden auf der Rotationsachse liegen; i. Allg. werden jedoch keine Körper erzeugt, weshalb im Folgenden der Begriff „Rotationsfläche“ verwendet wird.



Vielfältige Modellierungsmöglichkeiten ergeben sich durch die Rotation von Bézier- oder Splinekurven; Abb. 2.38 zeigt eine aus einer Catmull-Rom-Splinekurve (siehe S. 103) erzeugte Rotationsfläche (entsprechende farbige Darstellungen enthält S. 351). Analytische Beschreibungen sind zur Generierung derartiger Flächen nicht unbedingt erforderlich, da 3D-Grafikprogramme über Befehle verfügen, die Rotationsflächen durch die Eingabe von Stützpunkten erzeugen.<sup>62</sup>

Eine verallgemeinerte Vorgehensweise der Konstruktion von Flächen durch Kurven besteht darin, Kurven nicht nur entlang von Kurven zu extrudieren, sondern dabei in andere Kurven übergehen zu lassen. Dazu wird mit Querschnittskurven entlang der Extrusionskurve gearbeitet – diese können verändert oder durch andere Kurven ersetzt werden. Abb. 2.39 zeigt ein Bearbeitungsfenster des entsprechenden Werkzeugs („Spline Modeler“<sup>63</sup>) der 3D-Grafiksoftware Carrara.

Zur Konstruktion einer Fläche lassen sich in dieser Software beliebig viele Querschnittskurven einfügen und bearbeiten; außerdem ist eine Bearbeitung der Extrusionskurve (bzw. ihrer Projektionen auf die Koordinatenebenen) möglich.

Die Veränderung einzelner Punkte von Querschnittskurven verändert Bereiche der Fläche um die betreffenden Punkte. Insofern besitzt diese Vorgehensweise eine Ähnlichkeit zur Interpolation von Flächen durch gegebene Punkte, auf die im nächsten Abschnitt eingegangen wird.

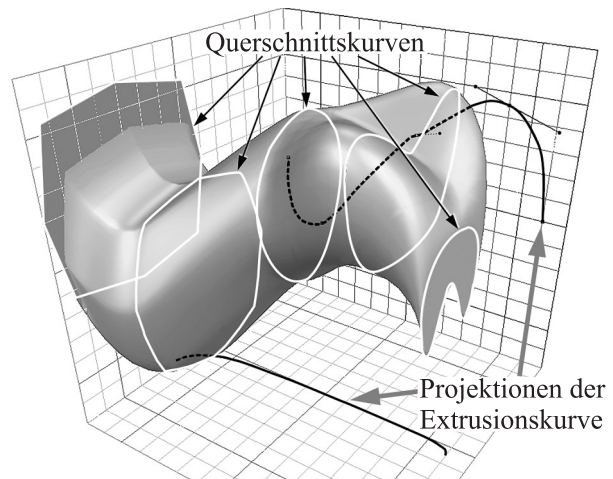


Abbildung 2.39: Spline-Modellierung

<sup>62</sup>In POV-Ray sind dies die Anweisungen `lathe`, womit Abb. 2.38 c) erzeugt wurde, und `sur` (Surface of Revolution), die Rotationskörper durch Bilden der konvexen Hülle der erzeugten Flächen darstellt.

<sup>63</sup>Der Spline Modeler von Carrara arbeitet mit Bézierkurven (siehe 2.3.6). Extrusions- und Querschnittskurven können durch Modifikation der Stützpunkte und Tangentenvektoren bearbeitet werden.

### 2.3.8 Freiformflächen

Als für die Modellierung von Formen extrem flexible Objekte haben in der Computergrafik und in der computergestützten Konstruktion Flächen eine zentrale Bedeutung erlangt, die durch beliebig im Raum angeordnete Punkte verlaufen oder sich diesen annähern. Die Form derartiger „Freiformflächen“ lässt sich durch die Veränderung von Gewichten und Tangenten sowie das Einfügen zusätzlicher Kontrollpunkte steuern und „verfeinern“.<sup>64</sup> Realisiert werden Flächen mit diesen Eigenschaften i. Allg. durch die mindestens einfach stetig differenzierbare (glatte) Zusammensetzung bikubischer Flächenstücke (Patches), also von Flächenstücken, die sich als Funktionen 3. Grades<sup>65</sup>

$$\vec{p}(u, v) = \sum_{i=0}^3 \sum_{j=0}^3 \vec{a}_{i,j} u^i v^j, \quad (2.10)$$

bzw. in Koordinatenschreibweise

$$x(u, v) = \sum_{i=0}^3 \sum_{j=0}^3 a_{i,j}^{(x)} u^i v^j, \quad y(u, v) = \sum_{i=0}^3 \sum_{j=0}^3 a_{i,j}^{(y)} u^i v^j, \quad z(u, v) = \sum_{i=0}^3 \sum_{j=0}^3 a_{i,j}^{(z)} u^i v^j \quad (2.11)$$

in zwei Parametern  $u$  und  $v$  darstellen lassen. Dabei sind  $\vec{a}_{i,j} = \left( a_{i,j}^{(x)}; a_{i,j}^{(y)}; a_{i,j}^{(z)} \right)^T$  (mit  $i, j = 0, \dots, 3$ ) die 16 Koeffizientenvektoren des Flächenstücks.<sup>66</sup>

Die in der Praxis gebräuchlichen Freiformflächen entstehen durch Erweiterung der in Abschnitt 2.3.6 beschriebenen Kurven zu *Tensorproduktflächen*. Diese können als „Kurven von Kurven“ aufgefasst werden. Werden alle Kontrollpunkte einer Bézier- oder Splinekurve entlang ebensolcher Kurven durch den Raum bewegt, so entstehen Kurvenscharen, die Flächen beschreiben. Dieses Konzept ist der am Ende von Abschnitt 2.3.7 auf S. 105 beschriebenen verallgemeinerten Extrusion von Kurven ähnlich; allerdings wird bei Tensorproduktflächen für jeden Kontrollpunkt eine Kurve (anstelle einer „globalen“ Extrusionskurve für die gesamte entstehende Fläche) betrachtet. Da Bézier- und Splinekurven Raumkurven sein können (in Abschnitt 2.3.6 wurden sie dimensionsunabhängig definiert) ist das Konzept der Tensorproduktflächen noch allgemeiner als das der verallgemeinerten Extrusionsflächen. Im Folgenden wird dieses Konzept exemplarisch anhand der Bézierflächen skizziert.

Die Tensorproduktfläche eines nach (2.4), Seite 99, durch  $\vec{p}(u) = \sum_{i=0}^3 B_{i,3}(u) \cdot \vec{p}_i$  beschriebenen kubischen Bézierkurvenstücks entsteht, indem die Ortsvektoren  $\vec{p}_i$  der Kontrollpunkte durch Bézierkurvenstücke ersetzt werden, die sich ebenfalls nach (2.4) durch  $\sum_{j=0}^3 B_{j,3}(v) \cdot \vec{p}_{i,j}$  ( $i = 0, \dots, 3$ ) beschreiben lassen. Für das entstehende Flächenstück ergibt sich daraus die Parameterdarstellung

<sup>64</sup>Praktische Aspekte der Modellierung mit Freiformflächen in 3D-Anwendungssoftware und entsprechende Beispiele werden u. a. in [45], S. 163–194; [220], S. 196–201 und [250], S. 107–176 behandelt.

<sup>65</sup>Die Verwendung von Funktionen 3. Grades hat sich in der Computergrafik aus den in Fußnote 50 auf S. 99 genannten Gründen durchgesetzt. Diese gelten für Flächen gleichermaßen.

<sup>66</sup>Im Folgenden wird nur die platz sparende Vektorschreibweise verwendet; die entsprechenden Koordinatendarstellungen ergeben sich daraus jeweils wie (2.11) aus (2.10).



$$\vec{p}(u, v) = \sum_{i=0}^3 B_{i,3}(u) \left( \sum_{j=0}^3 B_{j,3}(v) \cdot \vec{p}_{i,j} \right) = \sum_{i=0}^3 \sum_{j=0}^3 B_{i,3}(u) \cdot B_{j,3}(v) \cdot \vec{p}_{i,j} \quad (2.12)$$

mit den Bernsteinpolynomen

$$B_{i,3}(u) = \binom{3}{i} u^i (1-u)^{3-i} \quad \text{und} \quad B_{j,3}(v) = \binom{3}{j} v^j (1-v)^{3-j} \quad (u, v \in [0; 1]).$$

Die Form dieses Bézierflächenstücks (oft als *Bézierpatch* bezeichnet) wird durch die 16 Kontrollpunkte  $P_{i,j}$  mit den Ortsvektoren  $\vec{p}_{i,j}$  ( $i, j = 0, \dots, 3$ ) bestimmt. Davon gehören  $P_{0,0}$ ,  $P_{0,3}$ ,  $P_{3,0}$  und  $P_{3,3}$  dem Flächenstück an;  $P_{1,0}$ ,  $P_{0,1}$ ,  $P_{0,2}$ ,  $P_{1,3}$ ,  $P_{2,0}$ ,  $P_{3,1}$ ,  $P_{3,2}$  und  $P_{2,3}$  legen mit diesen zusammen die Tangentenvektoren der vier das Flächenstück begrenzenden Bézierkurvenstücke in ihren Endpunkten fest (siehe Abb. 2.40). Durch das Zusammenfügen von Bézierpatches mit jeweils 8 gemeinsamen Kontrollpunkten (analog zu der auf S. 99 beschriebenen Zusammensetzung von Kurvenstücken) lassen sich komplexe Flächen erstellen. So setzt sich das vielleicht bekannteste Objekt der Computergrafik, die Teekanne der Universität Utah, aus Bézierpatches zusammen (Abb. B.14 auf S. 351, siehe auch [257], S. 119).

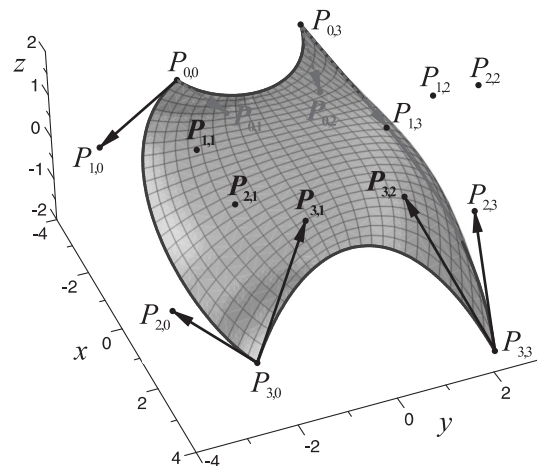


Abbildung 2.40: Bézierflächenstück

Eine farbige Version dieser Abbildung enthält S. 351. Dateien, mithilfe derer Bézierpatches in MuPAD erzeugt werden können, befinden sich in dem Ordner 2-3 (siehe S. 365).

In analoger Weise wie hier für Bézierpatches kurz dargestellt wurde, lassen sich die in Abschnitt 2.3.6 beschriebenen interpolierenden Spline-, B-Spline-, und NURBS-Kurven zu entsprechenden Tensorproduktflächen erweitern.<sup>67</sup>

Aufgrund der hohen Zahl einzugebender Kontrollpunkte ist die Modellierung mit Freiformkurven auf Koordinatenebene kaum möglich und bleibt deshalb Programmen mit grafischer (mausgestützter) Eingabe vorbehalten.<sup>68</sup> Aber auch die Nutzung der grafischen Bézier- und NURBS-Modellierungswerkzeuge in professioneller 3D-Grafik- und CAD-Software erfordert einen hohen Einarbeitungsaufwand. Die bereits für ein einzelnes Flächenstück recht große Anzahl von Kontrollpunkten (siehe Abb. 2.40) bedingt, dass viel Erfahrung notwendig ist, um die Wirkungen der Kontrollpunkte sowie von Gewichten und Tangenten auf die Flächenform für Modellierungszwecke zu nutzen.

<sup>67</sup>Die für die Kurven genannten Differenzierbarkeitseigenschaften gelten auch für die entsprechenden Flächen, siehe u. a. [6], S. 467–505; [31], S. 233–238; [79], S. 399–405 und [124], S. 277–293.

<sup>68</sup>POV-Ray ermöglicht die Erstellung bikubischer Flächen mittels der Anweisung `bicubic_patch`; die direkte Nutzung dieser Anweisung zu Modellierungszwecken ist jedoch kaum praktikabel. Es existieren hierfür grafische Eingabeprogramme, die entsprechenden POV-Ray-Code erzeugen (wie z. B. HamaPatch, siehe [www.geocities.com/hamapatch/index.html](http://www.geocities.com/hamapatch/index.html)).

### 2.3.9 Polygondarstellung von dreidimensionalen Objekten

#### Polygonnetze und Polyeder

3D-Grafikprogramme können nicht alle der in den vorangegangenen Abschnitten beschriebenen Darstellungsformen unmittelbar für die Bildberechnung (Rendering) verarbeiten. Intern wandeln sie daher in vielen Fällen Flächen bzw. Körper in Polygonnetze um.<sup>69</sup> Neben der durch Software intern vorgenommenen Umwandlung von geometrischen Objekten in Polygonnetze treten diese auch beim Datenaustausch zwischen verschiedenen Anwendungen sowie bei der Oberflächenerfassung von Objekten der Realität durch Abtastung mittels Laserstrahlen auf.

Unter einem *Polygonnetz* wird eine Menge  $M$  endlich vieler geschlossener, einfacher (und ebener<sup>70</sup>) Polygone ( $n$ -Ecke) mit folgenden Eigenschaften verstanden:

- Der Schnitt zweier Polygone aus  $M$  ist entweder leer, besteht aus einem Punkt oder besteht aus einer Kante.
- Jede Kante eines Polygons gehört zu höchstens einem weiteren Polygon.
- Die Menge aller Kanten, die nur zu einem einzigen Polygon gehören, ist leer oder bildet ein einziges geschlossenes Polygon.

Das Polygon, dessen Kanten alle diejenigen Kanten eines Polygonnetzes sind, die nur zu einem Polygon gehören, ist der *Rand* dieses Polygonnetzes (Abb. 2.41 b). Ein Polygonnetz ist *geschlossen*, wenn es keine Kanten enthält, die nur zu einem Polygon gehören. Ein geschlossenes Netz ebener Polygone bildet ein *Polyeder*, wenn die Vereinigungsmenge aller Polygone zusammenhängend ist und jeder Eckpunkt zu genau einer zyklisch geordneten Menge von Polygonen gehört, von denen je zwei aufeinander folgende eine gemeinsame Kante besitzen. Das Netz in Abb. 2.41 c) ist daher kein Polyeder.

Für die *Tessellation*, d. h. die Umwandlung durch Parameterdarstellungen gegebener Flächen (also auch der in Abschnitt 2.3.8 beschriebenen Freiformflächen) in Polygon-

<sup>69</sup>Viele Softwarepakete behandeln z. B. durch Gleichungen oder Parameterdarstellungen dargestellte Flächen als Polygonnetze und repräsentieren einfache Grundkörper als Polyeder (siehe S. 88 und 93).

<sup>70</sup>Die Forderung, dass ein Polygonnetz aus ebenen Polygonen bestehen muss, wird in den meisten Definitionen des Begriffs „Polygonnetz“ erhoben, jedoch nicht einheitlich gehandhabt. Oft werden auch Vierecksnetze, die Flächen durch das Verbinden von Punkten in zwei Richtungen durch Strecken approximieren, als Polygonnetze bezeichnet – obwohl die dabei auftretenden Vierecke i. Allg. nicht eben sind. Im Folgenden wird der Begriff „Polygonnetz“ auch in dieser allgemeineren Auffassung verwendet. Netze mit nicht ebenen Polygonen lassen sich durch das Einfügen zusätzlicher Kanten in ebenflächig begrenzte Polygonnetze (z. B. Dreiecksnetze) überführen, worauf noch eingegangen wird.

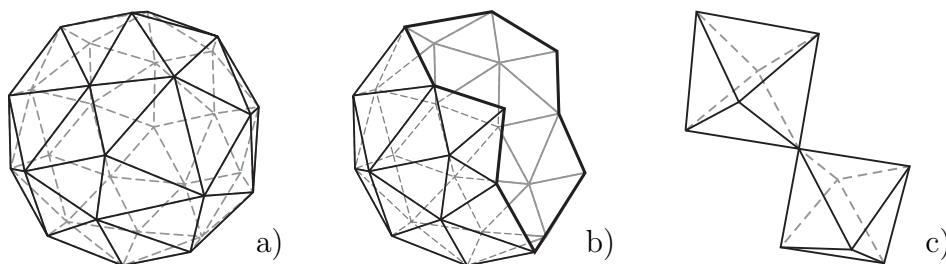


Abbildung 2.41:  
Polygonnetze

netze werden die Parameterintervalle  $[u_1; u_2]$  und  $[v_1; v_2]$ , auf denen die Flächen definiert sind, i. Allg. gleichmäßig unterteilt und für die Unterteilungspunkte  $(u_i; v_j)$  die Koordinaten der entsprechenden Flächenpunkte  $P(u_i; v_j)$  berechnet. Dadurch ergeben sich die Eckpunkte eines die jeweilige Fläche repräsentierenden Polygonnetzes,<sup>71</sup> das (von Ausnahmen bei singulären Punkten abgesehen) aus Vierecken besteht.

### Dreiecksnetze

Die Eckpunkte eines Polygons liegen meist nicht in einer Ebene. Da für die Bildberechnung in der Computergrafik Normalenvektoren von zentraler Bedeutung sind (siehe Abschnitt 2.5), werden Polygonnetze vor der Bildberechnung in *Dreiecksnetze* überführt, also durch Unterteilung von  $n$ -Ecken mit  $n \geq 4$  in Dreiecke „trianguliert“.

Dreieckige Facetten legen Ebenen fest; ihnen können daher Normalen zugeordnet werden, die für die Bildberechnung in der Computergrafik benötigt werden.

Mittels eines von MAEDER (siehe [171]) entwickelten Zusatzes wurde die in Abb. 2.26 d) auf S. 92 abgebildete (mit Mathematica dargestellte) konische Spirale in eine für die Weiterverarbeitung in POV-Ray geeignete Form als *mesh*-Objekt (Menge von Dreiecken) konvertiert. Sie wird dabei durch die Vereinigung von 792 Dreiecken mit folgender Syntax repräsentiert:

```
triangle { <x1,y1,z1> , <x2,y2,z2> , <x3,y3,z3> }.
```

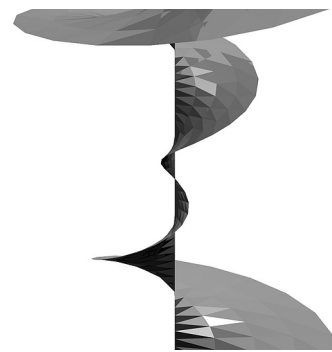


Abbildung 2.42: Dreiecksnetz

Abb. 2.42 zeigt das von POV-Ray berechnete Bild dieser Repräsentation.<sup>72</sup>

Die Beschreibung von Dreiecksnetzen durch Auflistung aller Dreiecke unter Angabe der Koordinaten der jeweiligen 3 Eckpunkte ist insofern uneffektiv, als jeder Eckpunkt mehreren Dreiecken angehört und deshalb mehrfach aufgelistet wird. Dies macht sich bei sehr detaillierten Dreiecksnetzen, bei denen die Koordinaten in hoher Genauigkeit angegeben sind, durch einen hohen Speicherbedarf und eine geringe Verarbeitungsgeschwindigkeit bemerkbar. Effizienter ist die Angabe einer Liste aller Eckpunkte (vertices) und die Beschreibung der Dreiecke durch die Indizes, die den entsprechenden Eckpunkten zugeordnet sind. Eine derartige Datenstruktur wird in POV-Ray als *mesh2* bezeichnet:

<code>mesh2 { vertex_vectors { n,</code>	Anzahl der Eckpunkte des Netzes
<code>&lt;x<sub>0</sub>,y<sub>0</sub>,z<sub>0</sub>&gt;, ...</code>	Koordinaten der Eckpunkte
<code>&lt;x<sub>n-1</sub>,y<sub>n-1</sub>,z<sub>n-1</sub>&gt; }</code>	mit den Indizes 0, ..., n-1
<code>face_indices { f,</code>	Anzahl der Dreiecke (Facetten, faces) des Netzes
<code>&lt;i,j,k&gt;, ...</code>	Angabe der $f$ Dreiecke durch die Indizes $i, j, k, l, m, o$
<code>&lt;l,m,o&gt; }</code>	der jeweiligen Eckpunkte ( $0 \leq i, j, k, l, m, o \leq n-1$ )

<sup>71</sup>Dieses Vorgehen ist bei der Flächendarstellung von CAS deutlich sichtbar, vgl. S. 91f. und S. 107.

<sup>72</sup>Die Ursache für die „harten“ Helligkeitsübergänge zwischen den Dreiecken besteht darin, dass die Schattierung (Ausleuchtung) von Facetten mittels ihrer Normalen erfolgt (siehe Abschnitt 2.5.2). Beim Übergang zwischen zwei Dreiecken ändern sich die Normalen abrupt. Interpolationsverfahren zur Glättung der dadurch verursachten „Helligkeitssprünge“ werden in Abschnitt 2.5.3 beschrieben.

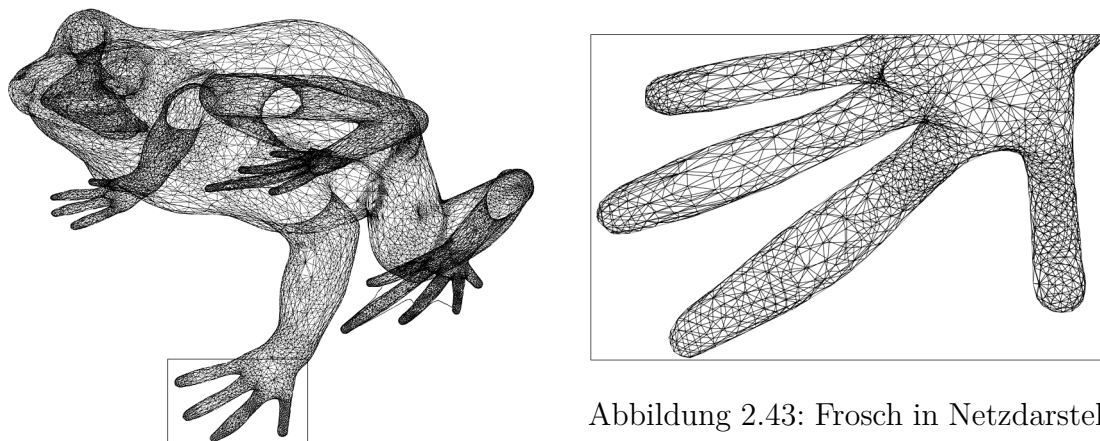


Abbildung 2.43: Frosch in Netzdarstellung

Wie bereits erwähnt wurde, sind Polygon- und speziell Dreiecksnetze nicht nur für die Umwandlung analytisch beschriebener Flächen zur weiteren Verarbeitung von Bedeutung, sondern vor allem für die Repräsentation „realer“ Objekte mit sehr vielen Facetten. Hierbei macht sich der Vorteil einer effizienten Beschreibung durch Listen deutlich bemerkbar. Die Dateigröße der **mesh**-Geometriebeschreibung des in Abb. 2.43 durch ein Dreiecksnetz dargestellten Frosches mit 25815 Eckpunkten und 48118 Dreiecken ist mit ca. 14,5 MB fast drei mal so groß wie die der **mesh2**-Beschreibung mit ca. 5,5 MB. Die Zeit, die POV-Ray für das Parsing (Erfassung der Geometriedaten) benötigte, betrug in der **mesh**-Darstellung 6 und in der **mesh2**-Repräsentation 2 Sekunden (Pentium IV; 3,2 GHz). Bei höher aufgelösten Netzen oder hohen Geschwindigkeitsanforderungen (z. B. bei Computerspielen) kann dieser Unterschied bedeutsam sein.

Da für Sichtbarkeits- und Schattierungsberechnungen Normalen(einheits-)vektoren der einzelnen Facetten von Dreiecksnetzen bestimmt werden müssen, sind die Reihenfolgen, in denen Eckpunkte von Dreiecken aufgezählt werden, von Bedeutung. Notwendig ist nämlich, dass alle Normalen eines Netzes in den – vom Netz aus betrachtet – selben Bereich des Raumes gerichtet sind (bei geschlossenen Netzen nach außen). Sind  $P_1$ ,  $P_2$  und  $P_3$  die Eckpunkte eines Dreiecks, so besitzt die Ebene durch dieses Dreieck genau zwei (zueinander entgegengerichtete) Normaleneinheitsvektoren:

$$\vec{n}_1 = \frac{\overrightarrow{P_1P_2} \times \overrightarrow{P_1P_3}}{|\overrightarrow{P_1P_2} \times \overrightarrow{P_1P_3}|} \quad \text{und} \quad \vec{n}_2 = -\vec{n}_1 = \frac{\overrightarrow{P_1P_3} \times \overrightarrow{P_1P_2}}{|\overrightarrow{P_1P_3} \times \overrightarrow{P_1P_2}|}.$$

Werden für alle Dreiecke eines Netzes Normalenvektoren jeweils durch das Vektorprodukt des Verbindungsvektors des ersten und des zweiten aufgezählten Punktes sowie des Verbindungsvektors des ersten und des dritten Punktes bestimmt, so hängt es von der Reihenfolge, in der die Punkte aufgelistet sind ab, welche Orientierung der berechnete Normalenvektor hat. Bereits bei der Generierung der Dreieckslisten muss daher eine *konsistente Orientierung* der Facetten gewährleistet werden, d. h. eine Reihenfolge der Eckpunkte eines jeden Dreiecks, bei der die einander entsprechenden Normalenvektoren jeweils zweier benachbarter Dreiecke in denselben Bereich des Raumes bezüglich des Netzes „zeigen“.

Diese Bedingung ist dann erfüllt, wenn in beliebigen benachbarten Dreiecken gemeinsame Kanten immer in jeweils entgegengesetzter Reihenfolge durchlaufen werden. Erfolgt also z. B. eine Auflistung der in Abb. 2.44 dargestellten Dreiecke mit den Punktreihenfolgen  $P_1P_2P_3$ ,  $P_1P_3P_4$ ,  $P_1P_4P_6$ ,  $P_6P_4P_5$  und  $P_6P_2P_1$  (wobei zyklische Vertauschungen möglich sind), so sind die Vektorprodukte  $\overrightarrow{AB} \times \overrightarrow{AC}$  (wobei  $A$  jeweils der erste,  $B$  der zweite und  $C$  der dritte aufgeführte Punkt ist) für alle Dreiecke in denselben Teil des Raumes bezüglich des Netzes gerichtet.

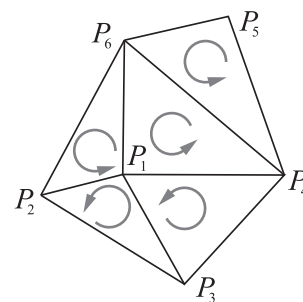


Abbildung 2.44: Orientierung von Facetten

### Subdivision Surfaces

Eine direkte Modellierung durch Eingabe der Eckpunktkoordinaten von Polygonnetzen (oder auch mausgestützt durch „Ziehen“ an Eckpunkten) ist nur für sehr einfache Objekte (wie z. B. platonische Körper) möglich.<sup>73</sup> Seit Ende der neunziger Jahre hat jedoch ein (bereits in den siebziger Jahren vorgeschlagenes) Modellierungsverfahren insbesondere für die Gestaltung „weicher“ Formen Verbreitung gefunden, bei dem grobe Polygonnetze durch Manipulation von Ecken geformt und daraus mithilfe mehrfacher Unterteilungen feine Netze mit kaum noch wahrnehmbaren Ecken und Kanten generiert werden. Dieses als *Subdivision Modeling* bezeichnete Verfahren sei hier kurz für den (einfacheren) Fall ebener Polygonzüge beschrieben.

Für drei aufeinander folgende Ecken  $P_{j-1}$ ,  $P_j$ ,  $P_{j+1}$  eines Polygonzuges seien  $M_j$  und  $M_{j+1}$  die Mittelpunkte der Kanten  $\overline{P_{j-1}P_j}$  und  $\overline{P_jP_{j+1}}$ . Die Kanten des Polygonzuges  $P_{j-1}M_jP_jM_{j+1}P_{j+1}$  werden wiederum halbiert und die dabei entstandenen Mittelpunkte durch Strecken verbunden, deren Mittelpunkte die Ecken  $P_i^{\text{neu}}$  des „neuen“ Polygonzuges sind (siehe Abb. 2.45 a). Indiziert man die ursprünglichen Ecken und Kantenmittelpunkte mit  $P_{i-2} := P_{j-1}$ ,  $P_{i-1} := M_j$ ,  $P_i := P_j$ ,  $P_{i+1} := M_{j+1}$  und  $P_{i+2} := P_{j+1}$ , so gilt

$$\begin{aligned} \vec{p}_i^{\text{neu}} &= \frac{\frac{\vec{p}_{i-1} + \vec{p}_i}{2} + \frac{\vec{p}_i + \vec{p}_{i+1}}{2}}{2} \\ &= \frac{\vec{p}_{i-1}}{4} + \frac{\vec{p}_i}{2} + \frac{\vec{p}_{i+1}}{4} \end{aligned}$$

für alle Punkte  $P_i^{\text{neu}}$  außer dem ersten und dem letzten Punkt des Netzes (diese bleiben unverändert). Der unterteilte Streckenzug kann nach demselben Verfahren erneut unterteilt werden. Die Verfeinerung eines Polygonzuges nach zweifacher Unterteilung zeigt Abb. 2.45 b).

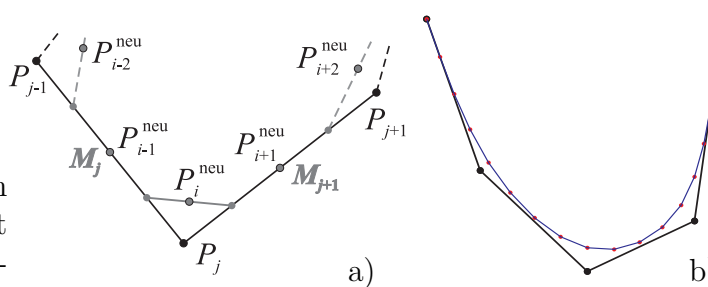


Abbildung 2.45: Unterteilung eines Polygonzuges

Die Datei, mit der Abb. 2.45 b) in MuPAD generiert wurde, befindet sich in dem Ordner 2-3 (siehe S. 365). Mithilfe dieser Datei ist es möglich, das beschriebene Unterteilungsverfahren durch Wahl einer Iterationstiefe beliebig oft auf einen Polygonzug anzuwenden.

<sup>73</sup>In Abschnitt 4.6.6 wird eine Unterrichtseinheit beschrieben, in der Schüler ein Oktaeder durch seine dreieckigen Facetten darstellen.

Das beschriebene Verfahren der Verfeinerung von Polygonzügen lässt sich auf Polygonnetze im Raum übertragen, indem anstelle von Streckenmittelpunkten Schwerpunkte der Polygone, welche die Facetten des Netzes bilden, bestimmt und mit diesen als Eckpunkte neue Polygone gebildet werden (siehe [11], S. 223–232). Abb. 2.46 zeigt das Modellierungswerkzeug „Vertex Modeler“ der Software Carrara, welches nach diesem Verfahren arbeitet. Dabei wurde eine fünffache Unterteilung gewählt. Durch Ziehen an den Eckpunkten des abgebildeten Steuernetzes lässt sich das glatt erscheinende unterteilte Netz interaktiv bearbeiten.

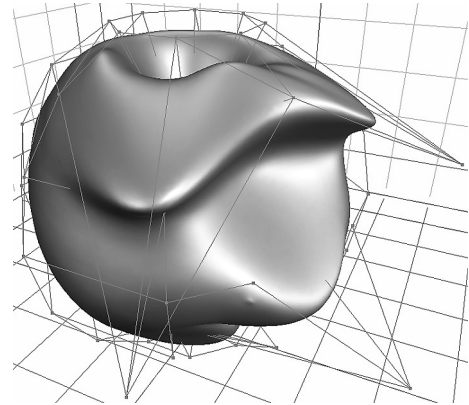


Abbildung 2.46: Subdivision Surface mit fünffacher Unterteilung

Die Verarbeitung von Subdivision Surfaces stellt hohe Anforderungen an die Hardware und wird daher für die Modellierung „organisch“ wirkender Objekte, i. Allg. jedoch nicht für die Glättung sichtbarer Kanten zwischen Facetten (wie in Abb. 2.42) eingesetzt.<sup>74</sup>

### 2.3.10 Diskretisierung des Raumes (Voxel)

Die Diskretisierung des Raumes unterscheidet sich in der Herangehensweise grundsätzlich von der analytischen Beschreibung einzelner Objekte und ihrer Annäherung durch Polygonnetze. Der für die Erzeugung einer Abbildung relevante Teil des Raumes wird hierbei in kleine Würfel unterteilt; diese Elementarwürfel werden als *Voxel* bezeichnet (siehe u. a. [125], S. 219–221; [248], S. 62–74 sowie [257], S. 66–72 und S. 413–434).<sup>75</sup> Zu den wichtigsten Anwendungen der Voxel-Unterteilung des Raumes zählt die medizinische Bildverarbeitung, insbesondere die Tomographie. Aufzeichnungsgeräte (vor allem Röntgengeräte) erfassen dabei zweidimensionale Schichtbilder eines menschlichen Körpers, die dann zu einer 3D-Repräsentation „zusammengesetzt“ werden. Somit wird eine dreidimensionale Darstellung der Transparenz<sup>76</sup> des menschlichen Körpers erzeugt.

Im Gegensatz zu den in den vorangegangenen Abschnitten beschriebenen Objektrepräsentationen durch Oberflächen werden bei der Diskretisierung von Teilen des Raumes durch Voxel Volumeneinheiten erfasst. Derartige volumenbasierte Arbeitsweisen verursachen extrem hohe Datenmengen und werden deshalb nur für spezielle Anwendungen eingesetzt. Neben der Medizin zählt hierzu die computerbasierte Strömungsdynamik. In der großen Mehrzahl der Einsatzgebiete der 3D-Computergrafik kommen oberflächenbasierte Repräsentationen dreidimensionaler Objekte zur Anwendung.<sup>77</sup> Da die mathematischen Grundlagen der Diskretisierung des Raumes zudem weniger in der analytischen Geometrie als vielmehr in einer effizienten Hierarchisierung, Datenrepräsentation und -verarbeitung liegen, wird hier auf eine nähere Darstellung verzichtet.

<sup>74</sup>Verfahren zur Glättung harter Farbübergänge an Kanten werden in Abschnitt 2.5.3 beschrieben.

<sup>75</sup>Voxel bilden das räumliche Analogon zu den Pixeln, durch die Bilder beschrieben werden, vgl. 2.1.1.

<sup>76</sup>Mit Transparenz ist dabei die Röntgendurchlässigkeit der Volumeneinheiten des Körpers gemeint.

<sup>77</sup>Für die Erzeugung von volumetrischen Spezialeffekten wie z. B. Rauch und Feuer kommen allerdings mitunter Algorithmen zur Anwendung, die auf Voxelbeschreibungen basieren.

## 2.4 Transformationen, Sichtbarkeitsuntersuchungen

Geometrische Transformationen und ihre Beschreibung, vor allem durch Matrizen, sind in der 3D-Computergrafik in mehreren Zusammenhängen von Bedeutung:

- Oft ist es sinnvoll, Positionen und Größen geometrischer Objekte nach ihrer Erstellung zu verändern, da die Wahl einer speziellen Lage die Beschreibung durch Koordinaten oder Gleichungen häufig vereinfacht. Durch Transformationen können die Objekte dann in die gewünschte Lage und Größe gebracht werden.
- Für die Erstellung von Animationen (Filmen) müssen vielfältige zeitabhängige Transformationen durchgeführt werden (siehe Abschnitt 2.6).
- Zur letztendlichen Bilderzeugung erfolgt die Projektion des dreidimensionalen Raumes in eine Ebene, wobei Zentralprojektionen, in Ausnahmefällen Parallelprojektionen, vorgenommen werden.
- Um Objekte mit Oberflächen zu versehen (zu texturieren), werden oft Bilder auf die Objekte projiziert (Abschnitt 2.4.6).

### 2.4.1 Affine Abbildungen des Raumes auf sich

Während affine Abbildungen jedem Punkt des Raumes einen Bildpunkt zuordnen, interessieren in der Computergrafik oftmals nur die Bildpunkte eines einzelnen Objekts, auf das eine Transformation angewendet wird. Objekte können in 3D-Grafiksoftware u. a. verschoben, gedreht, gespiegelt und skaliert (gestreckt bzw. gestaucht) werden.

Abb. 2.47 (rechts) zeigt die Wirkung dieser Transformationen anhand eines Torus, der links in seiner Ausgangslage abgebildet ist:

```
torus { 1, 0.2
  scale 1.5
  rotate <60,30,0>
  translate <-0.5,0.5,0.5> }
```

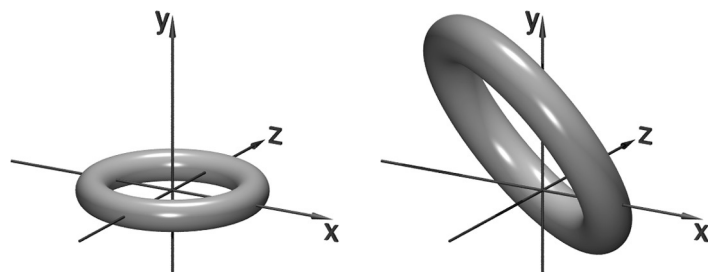


Abbildung 2.47: Transformation eines Torus

Die drei Argumente in **rotate**  $\langle\phi_x, \phi_y, \phi_z\rangle$  geben die Drehwinkel um die Koordinatenachsen, die Argumente von **translate** die Komponenten des Verschiebungsvektors und das Argument von **scale** den Streckungsfaktor einer zentrischen Streckung mit dem Koordinatenursprung als Zentrum an. Es ist auch möglich, Skalierungen mit unterschiedlichen Streckungsfaktoren in den Richtungen der Koordinatenachsen vorzunehmen: **scale**  $\langle s_x, s_y, s_z \rangle$ ; dabei werden Objekte entsprechend verzerrt.

Bekanntermaßen ist die Nacheinanderausführung von Transformationen i. Allg. nicht kommutativ.<sup>78</sup> Die Vertauschung der Reihenfolge von Transformationen führt daher zu anderen Positionen von Bildobjekten.

<sup>78</sup>Drehungen erfolgen in Grafikprogrammen stets um die Koordinatenachsen eines festgelegten Koordinatensystems; bei Streckungen liegt das Streckzentrum im Koordinatenursprung.

Neben der elementaren Beschreibung von Transformationen durch Anweisungen wie **translate**, **rotate**, **scale** lässt sich auch die universellere Matrizendarstellung nutzen, welche Grafiksoftware intern ohnehin für die Berechnung der Bildkoordinaten verwendet. Affine Abbildungen des Raumes können durch Abbildungsgleichungen

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} \quad (2.13)$$

dargestellt werden. Dabei sind  $x, y, z$  die Urbild- und  $x', y', z'$  die Bildkoordinaten beliebiger Punkte,  $\begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}$  der Translationsvektor sowie  $\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$  die Matrix der zugehörigen linearen Abbildung. Alternativ zu dieser Darstellung lassen sich affine Abbildungen durch homogene Matrizen der Form

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = A \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad \text{mit} \quad A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & t_x \\ a_{21} & a_{22} & a_{23} & t_y \\ a_{31} & a_{32} & a_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.14)$$

beschreiben, wobei Punkte durch homogene Koordinaten (siehe S. 86) anzugeben sind. Die Verwendung homogener Transformationsmatrizen hat den Vorteil, dass die Nacheinanderausführung beliebiger affiner Abbildungen durch das Produkt der zugehörigen Matrizen repräsentiert wird. Grafikprogramme führen Berechnungen daher mit homogenen Matrizen aus. In POV-Ray können Transformationsmatrizen mittels

`matrix < a11,a21,a31 , a12,a22,a32 , a13,a23,a33 , tx,ty,tz >`

auf Objekte angewendet werden.

*Bewegungen* lassen sich als Nacheinanderausführungen von *Drehungen*, *Translationen* und *Spiegelungen* allgemein durch Transformationsgleichungen der Form

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} + \begin{pmatrix} \cos \phi_z & -\sin \phi_z & 0 \\ \sin \phi_z & \cos \phi_z & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \cos \phi_y & 0 & -\sin \phi_y \\ 0 & 1 & 0 \\ \sin \phi_y & 0 & \cos \phi_y \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \phi_x & -\sin \phi_x \\ 0 & \sin \phi_x & \cos \phi_x \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \pm 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

oder durch homogene Matrizen der Form

$$A = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \cos \phi_z & -\sin \phi_z & 0 & 0 \\ \sin \phi_z & \cos \phi_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \cos \phi_y & 0 & -\sin \phi_y & 0 \\ 0 & 1 & 0 & 0 \\ \sin \phi_y & 0 & \cos \phi_y & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \phi_x & -\sin \phi_x & 0 \\ 0 & \sin \phi_x & \cos \phi_x & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \pm 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



bzw.

$$A = \begin{pmatrix} \cos \phi_y \cos \phi_z & -\cos \phi_z \sin \phi_x \sin \phi_y & \mp \cos \phi_z \cos \phi_x \sin \phi_y & t_x \\ \cos \phi_y \sin \phi_z & \cos \phi_z \cos \phi_x & \mp \cos \phi_z \sin \phi_x & t_y \\ \sin \phi_y & -\sin \phi_x \sin \phi_y \sin \phi_z & \mp \cos \phi_x \sin \phi_y \sin \phi_z & t_z \\ 0 & 0 & \pm \cos \phi_x \cos \phi_y & 1 \end{pmatrix}$$

darstellen. Dabei sind  $\phi_x$ ,  $\phi_y$  und  $\phi_z$  die Drehwinkel um die Koordinatenachsen (Eulersche Winkel). Wird an den Stellen, an denen „ $\pm$ “ auftritt, „+“ eingesetzt, so handelt es sich um eigentliche (orientierungserhaltende), für „-“ um uneigentliche Bewegungen.

*Streckungen* (oft auch als *Skalierungen* bezeichnet) werden durch Matrizen der Form<sup>79</sup>

$\begin{pmatrix} k_x & 0 & 0 & 0 \\ 0 & k_y & 0 & 0 \\ 0 & 0 & k_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$  beschrieben, wobei  $k_x$ ,  $k_y$  und  $k_z$  die Streckungsfaktoren entlang der drei Achsen sind; für  $k_x = k_y = k_z$  ergeben sich zentrische Streckungen.

Neben Bewegungen und Streckungen werden in der Computergrafik mitunter auch *Scherungen in den Richtungen der Koordinatenachsen* durchgeführt.

Die durch die Matrix  $\begin{pmatrix} 1 & 0,7 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$  beschriebene Abbil-

dung bewirkt eine Scherung entlang der  $x$ -Achse um 0,7 für jede Einheit in  $y$ -Richtung. Abb. 2.48 a), erzeugt durch

`cylinder { <0,-1.2,0> <0,1.2,0> 1.5`  
`matrix <1,0,0, 0.7,1,0, 0,0,1, 0,0,0> },`

zeigt die Wirkung dieser Scherung auf einen Zylinder. Bei

Abb. 2.48 b) wurde die Matrix  $\begin{pmatrix} 1 & 0,5 & 0,5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$  auf einen

Würfel angewendet, der dadurch entlang der  $x$ -Achse um 0,5 für jede Einheit in  $y$ -Richtung und für jede Einheit in  $z$ -Richtung geschert wird.

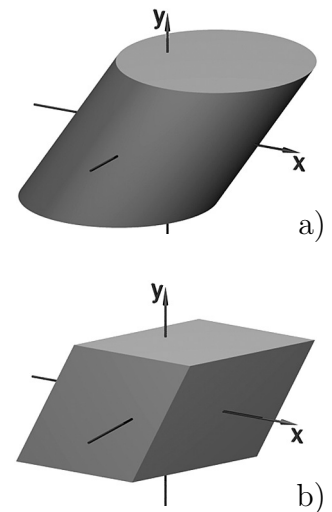


Abbildung 2.48: Scherungen entlang der  $x$ -Achse

Es existieren keine grundsätzlich anderen bijektiven affinen Abbildungen als die hier betrachteten – jede affine Abbildung des Raumes auf sich lässt sich als Nacheinanderausführung von Bewegungen, Streckungen und Scherungen darstellen. Mitunter wird der Begriff „Scherung“ auch in einem allgemeineren Sinne, Bewegungen und Streckungen beinhaltend, aufgefasst und synonym zu dem Begriff „affine Abbildung des Raumes auf sich“ verwendet.

<sup>79</sup>Im Folgenden werden nur die homogenen Matrizen angegeben.

## 2.4.2 Koordinatentransformationen

Bei der Arbeit mit Szenen in der 3D-Computergrafik treten mehrere Koordinatensysteme und Übergänge zwischen diesen auf:<sup>80</sup>

- Jede Szene wird bezüglich eines globalen Koordinatensystems, des *Szenen-* bzw. *Weltkoordinatensystems* konstruiert. Alle Objekte müssen durch Koordinaten bezüglich dieses Koordinatensystems beschrieben bzw. ihre Koordinaten müssen in Koordinaten bezüglich des Szenenkoordinatensystems überführt werden.
- Wie auf S. 113 ausgeführt wurde, verwenden 3D-Grafikprogramme als Fixpunkt bei Drehungen und Skalierungen stets den Koordinatenursprung. Da es bei komplexeren Transformationen schwierig sein kann, den Überblick über deren Wirkung zu behalten, ermöglichen viele Programme die Arbeit mit *Objektkoordinatensystemen* (lokalen Koordinatensystemen), bezüglich derer Objekte in einer günstigen Lage angeordnet und gedreht sowie skaliert werden können. Durch eine Koordinatentransformation werden vor der Berechnung des Bildes die Objektkoordinaten in Koordinaten bezüglich des Szenenkoordinatensystems überführt.
- Vor der Abbildung der dreidimensionalen Szene in eine Ebene erfolgt eine Transformation der Szenen- in Ansichtskordinaten. Dabei handelt es sich um Koordinaten bezüglich eines *Kamera-* bzw. *Ansichtskordinatensystems*, welches an der Projektionsebene und dem Projektionszentrum (bei einer Zentralprojektion) bzw. der Projektionsrichtung (bei einer Parallelprojektion) ausgerichtet ist.

Als Beispiel zeigt Abb. 2.49 a) ein Objektkoordinatensystem für eine bezüglich dieses Koordinatensystems mittels `scale <4,1.2,2>` ungleichmäßig skalierte Kugel, das Szenen- bzw. Weltkoordinatensystem sowie das Ansichtskordinatensystem für die Kamera, mit der die Grafik in Abb. 2.49 b) „aufgenommen“ wurde.

<sup>80</sup>Die Übergänge zwischen den beschriebenen Koordinatensystemen werden – in Verbindung mit der Projektion in eine Ebene (2.4.3) sowie Sichtbarkeitsuntersuchungen (2.4.4, 2.4.5) – häufig als geometrischer Teil einer „Computergrafik-Pipeline“ bezeichnet. Dabei handelt es sich um die Abfolge aller Arbeitsschritte von der Objektmodellierung bis hin zur Abbildung auf dem Monitor (siehe S. 84 sowie [11], S. 71f. und [257], S. 165ff.).

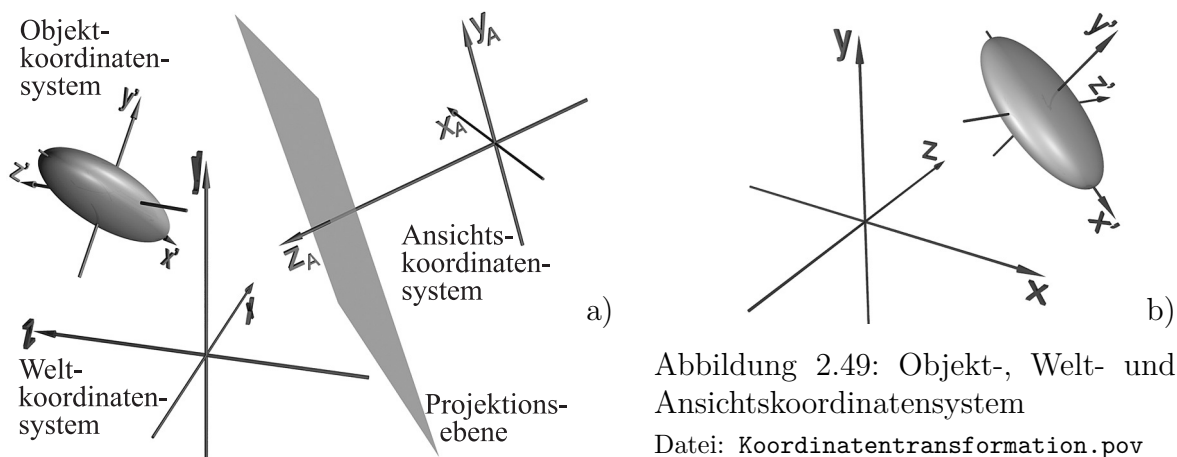


Abbildung 2.49: Objekt-, Welt- und Ansichtskordinatensystem

Datei: Koordinatentransformation.pov

Koordinatentransformationen lassen sich wie affine Abbildungen durch homogene Matrizen der Form (2.14), S. 114, beschreiben. Sind  $K = (O, \vec{b}_1, \vec{b}_2, \vec{b}_3)$  und  $K' = (O', \vec{b}'_1, \vec{b}'_2, \vec{b}'_3)$  zwei durch Ursprungspunkte und Basisvektoren gegebene Koordinatensysteme, so ergeben sich die Koeffizienten  $a_{i,j}$  ( $1 \leq i, j \leq 3$ ) der Matrix  $A$  in (2.14) durch Darstellung der Basisvektoren  $\vec{b}'_1, \vec{b}'_2, \vec{b}'_3$  als Linearkombinationen von  $\vec{b}_1, \vec{b}_2, \vec{b}_3$ :

$$\vec{b}'_j = \sum_{i=1}^3 a_{i,j} \vec{b}_i \quad (1 \leq j \leq 3).$$

Der Translationsvektor  $\begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}$  überführt den Koordinatenursprung  $O$  in  $O'$ ; es gilt also

$$O' = O + \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}$$

und somit  $t_x = x_{O'}$ ,  $t_y = y_{O'}$  und  $t_z = z_{O'}$ , wobei  $x_{O'}, y_{O'}, z_{O'}$  die Koordinaten von  $O'$  bezüglich  $K$  sind.

Die Matrix  $A$ , die durch Einsetzen der Koeffizienten  $a_{i,j}$  und  $t_x, t_y, t_z$  in (2.14) entsteht, überführt die Basisvektoren von  $K$  in diejenigen von  $K'$  sowie  $O$  in  $O'$ . Für die Koordinaten  $x, y, z$  und  $x', y', z'$  bezüglich  $K$  bzw.  $K'$  eines beliebigen Punktes gilt

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = A \cdot \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & t_x \\ a_{21} & a_{22} & a_{23} & t_y \\ a_{31} & a_{32} & a_{33} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} \quad \text{sowie} \quad \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = A^{-1} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}.$$

Die Koordinaten werden durch die Transformationsmatrix im Vergleich zu den Koordinatensystemen also in umgekehrter Richtung überführt. Dies lässt sich anhand von Abb. 2.49 veranschaulichen. Der Koordinatenursprung  $O'$  des Objektkoordinatensystems  $K'$  sei das Bild des Koordinatenursprungs  $O$  des Weltkoordinatensystems  $K$  bei Anwendung einer durch eine Matrix  $A$  beschriebenen affinen Abbildung. Dann ergeben sich die Koordinaten von  $O'$  bezüglich  $K$  durch Multiplikation seines Koordinatenvektors  $(0; 0; 0; 1)^T$  bezüglich  $K'$  mit  $A$ . Dies gilt in analoger Weise für die Koordinaten beliebiger Punkte bezüglich  $K$  und  $K'$ .

Affine Abbildungen und Koordinatentransformationen lassen sich als äquivalent zueinander auffassen. So führt die Beschreibung von Objekten in einem Objektkoordinatensystem bei anschließender Transformation der Koordinaten in die des Weltkoordinatensystems zu demselben Ergebnis wie die Modellierung in einer ausgezeichneten Lage bezüglich des Weltkoordinatensystems und anschließende Abbildung in die gewünschte Lage (wobei in beiden Fällen jeweils zueinander inverse Matrizen zu verwenden sind). Bei der Erstellung von Abb. 2.49 in POV-Ray musste der zweite Weg gewählt werden, da diese Software die Nutzung von Objektkoordinatensystemen nicht unmittelbar unterstützt, aber mit dem beschriebenen „Umweg“ ein vergleichbares Arbeiten ermöglicht.

### 2.4.3 Abbildungen in eine Ebene (Projektionen)

Um aus einer dreidimensionalen Szene ein Bild zu erzeugen, muss die Szene in eine Ebene projiziert werden. Dabei werden – entsprechend der in Abschnitt 2.2 beschriebenen Analogie zwischen 3D-Computergrafik und Fotografie – meist Zentralprojektionen, mitunter aber auch Parallelprojektionen durchgeführt.

*Parallelprojektionen* sind wegen des geringeren Rechenaufwands vor allem für Echtzeit-3D-Darstellungen von Bedeutung. Es erfolgt dazu eine Transformation des Szenen- bzw. Weltkoordinatensystems in ein Ansichtskordinatensystem (siehe S. 116), so dass die Projektionsebene mit der  $x$ - $y$ -Ebene übereinstimmt. Die Bildkoordinaten  $(x'; y'; z'; 1)$  beliebiger Punkte  $P(x; y; z; 1)$  ergeben sich dann sehr einfach durch  $x' = x$ ,  $y' = y$  und  $z' = 0$ . Die  $z$ -Koordinaten geben die „Tiefen“ von Punkten an und entscheiden bei Punkten mit gleichen  $x$ - und  $y$ -Koordinaten, welche Punkte verdeckt sind.

Bei einer *Zentralprojektion* mit dem Projektionszentrum  $Z$  und der Projektionsebene  $\varepsilon$  wird jeder Punkt  $P$  des Raumes auf den Schnittpunkt  $P'$  der Geraden  $ZP$  mit der Projektionsebene abgebildet (Abb. 2.50). Auch bei Zentralprojektionen wird vor der eigentlichen Projektion i. Allg. zunächst eine Ansichtstransformation, d. h. eine Koordinatentransformation vorgenommen, in deren Ergebnis die Projektionsebene parallel zur  $x$ - $y$ -Ebene des (Ansichts-)Koordinatensystems ist und das Projektionszentrum in dessen Ursprung liegt.

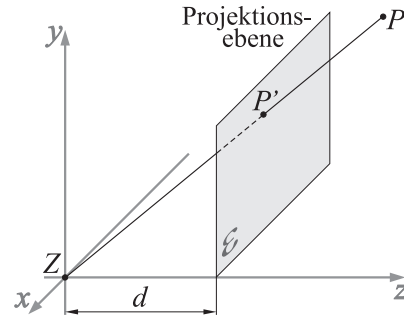


Abbildung 2.50: Zentralprojektion

Nach dem Strahlensatz ergibt sich für die Koordinaten  $x'$ ,  $y'$  des Bildpunktes  $P'$  eines Punktes  $P(x; y; z)$ :

$$x' = \frac{d \cdot x}{z} \quad \text{und} \quad y' = \frac{d \cdot y}{z},$$

wobei  $d$  der Abstand des Projektionszentrums von der Projektionsebene ist. Besonders elegant lässt sich die Zentralprojektion in homogenen Koordinaten darstellen; hierbei gilt für die Bildkoordinaten eines (vorher normierten) Punktes  $P(x; y; z; 1)$ :

$$(x'; y'; z'; w') = \left(x; y; z; \frac{z}{d}\right) \quad \text{bzw.} \quad \begin{pmatrix} x' \\ y' \\ z' \\ w' \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}.$$

Während die Bildberechnung durch geeignete Software ausgeführt wird, ist die Wahl von Projektionszentrum und -ebene durch den Benutzer vorzunehmen. In 3D-Grafiksoftware sind nicht Projektionszentrum und -ebene anzugeben, sondern die Position der Kamera (in `POV-Ray location`, entspricht dem Projektionszentrum  $Z$ ) sowie ein Vektor  $\vec{d}$ , der ihre Richtung beschreibt (`direction`) oder ein Zielpunkt  $L$ , auf den die Kamera gerichtet ist (`look_at`), siehe Abb. 2.51.

Die Projektionsebene ist die zu dem **direction**-Vektor  $\vec{d}$  (bzw. dem dazu kollinearen Vektor  $\overrightarrow{ZL}$ ) orthogonale Ebene durch den Zielpunkt  $L$ . Da bei computergrafischen Darstellungen nicht der gesamte Raum projiziert, sondern lediglich ein Teil des Raumes auf einen rechteckigen Ausschnitt der Projektionsebene – das *Sichtfenster* – abgebildet werden soll, müssen zusätzlich Angaben zum Bildausschnitt (**up** und **right**) oder zum Öffnungswinkel der Kamera (**angle**) gemacht werden, durch die dann die Größe des Bildausschnittes festgelegt wird. Wird statt **up** und **right** nur der Öffnungswinkel angegeben, so verwendet POV-Ray als Vertikale (**up**) einen Vektor, der sich als Linearkombination von  $\vec{d}$  und dem Richtungsvektor der  $y$ -Achse des Szenenkoordinatensystems darstellen lässt und zu  $\vec{d}$  orthogonal ist.

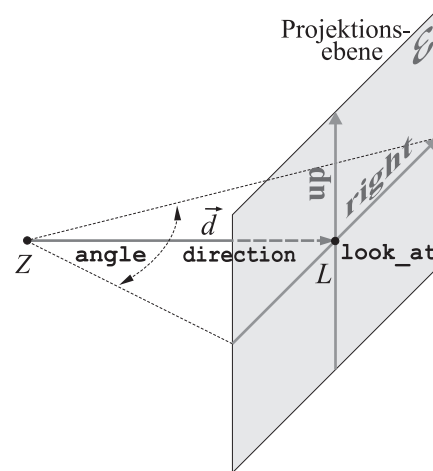


Abbildung 2.51: Beschreibung einer Zentralprojektion durch Angaben zur Kamera

Bei der oft genutzten Kamerabeschreibung durch **location**, **look\_at** und **angle** entspricht somit die  $y$ -Richtung des Koordinatensystems, bezüglich dessen die Beschreibung der Szene erfolgt, der vertikalen Richtung des erzeugten Bildes (siehe Abb. 2.49).

#### 2.4.4 Clipping von Objekten außerhalb des Sichtvolumens

Da bei computergrafischen Darstellungen nur ein Teil des Raumes auf einen i. Allg. rechteckigen Ausschnitt der Projektionsebene abgebildet wird, ist es im Hinblick auf die Geschwindigkeit der Bildberechnung sinnvoll, nicht sichtbare Objekte von weiteren Berechnungen auszuschließen.<sup>81</sup> Dazu ist das so genannte Sichtvolumen zu beschreiben, d. h. die Menge aller Punkte des Raumes, die bei einer Parallel- bzw. Zentralprojektion auf das Sichtfenster abgebildet werden.<sup>82</sup> Das Sichtvolumen ist bei einer Zentralprojektion eine „unendliche“ Pyramide (siehe Abb. 2.52 auf S. 120). Bei praktisch angewendeten Echtzeit-3D-Darstellungen werden häufig nur Objekte abgebildet, die sich, vom Sichtpunkt bzw. der Kamera aus betrachtet, „hinter“ dem Sichtfenster befinden und außerdem eine (i. Allg. recht hoch festgelegte) maximale Entfernung zum Sichtpunkt nicht überschreiten. Das Sichtvolumen reduziert sich damit auf einen Pyramidenstumpf bei Zentralprojektionen bzw. einen Quader bei Parallelprojektionen.

Nach der Bestimmung des Sichtvolumens können alle Objekte, die sich vollständig außerhalb desselben befinden, aus der Objektliste entfernt werden. Die Entscheidung,

<sup>81</sup>Für die fotorealistische 3D-Computergrafik, bei der Spiegelungen der Umgebung in den Oberflächen sichtbarer Objekte auftreten sollen, müssen jedoch auch Objekte berücksichtigt werden, die nicht direkt sichtbar sind. Somit sind für Verfahren, bei denen Fotorealismus im Vordergrund steht (wie Raytracing, siehe 2.5.5), die hier dargestellten Überlegungen zum Clipping gegenstandslos.

<sup>82</sup>Objekte, für die das Projektionszentrum zwischen Objekt und Sichtfenster liegt, sind jedoch nicht zu berücksichtigen, da „hinter“ einer Kamera befindliche Objekte natürlich nicht fotografiert werden.

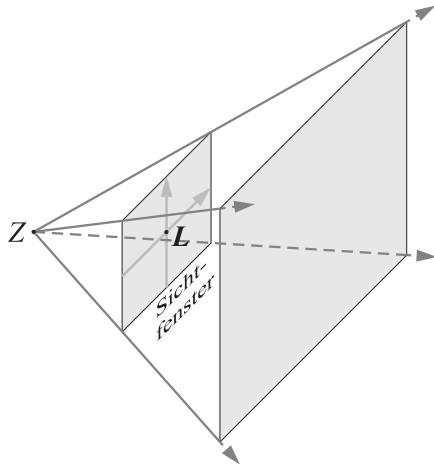


Abbildung 2.52: Sichtvolumen

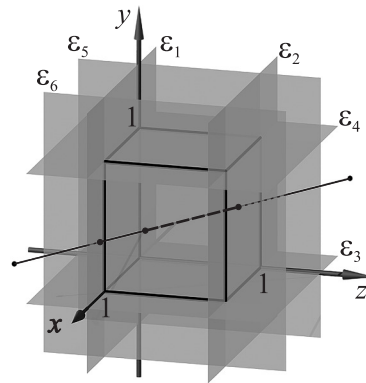


Abbildung 2.53: Clipping-Ebenen

ob Objekte außerhalb des Sichtvolumens liegen, ist für Quader einfacher zu treffen als für Pyramidenstümpfe. Daher wird bei Zentralprojektionen das Sichtvolumen mittels einer projektiven Transformation, die das Projektionszentrum auf einen uneigentlichen Punkt (vgl. S. 86) abbildet, in einen Quader überführt (siehe u. a. [11], S. 51–55 und [248], S. 36–38). Anschließend wird der Sichtquader mittels einer affinen Transformation normiert, d. h. in den Würfel überführt, dessen sämtliche Eckpunktkoordinaten 0 oder 1 sind. Beide Transformationen sind auf alle Objekte der Szene anzuwenden.

Da Clipping-Algorithmen vor allem für „schnelle“ 3D-Anwendungen (bei denen Objekte als Polygonnetze repräsentiert werden) von Interesse sind, werden diese i. Allg. an Strecken und Polygonen ausgeführt. Um den innerhalb des Sichtquaders liegenden Teil einer Strecke zu ermitteln, ist zunächst für ihre Eckpunkte festzustellen, ob sie (aus Sicht des Quaders) „vor“ oder „hinter“ einer der sechs Clipping-Ebenen (durch die Seitenflächen des Quaders bestimmte Ebenen, siehe Abb. 2.53) liegen. Gibt es eine Clipping-Ebene, hinter der beide Endpunkte liegen, so ist die Strecke vollständig unsichtbar; liegen beide Eckpunkte vor allen dieser Ebenen, so ist sie vollständig sichtbar.<sup>83</sup> In allen anderen Fällen müssen die Schnittpunkte der Strecke mit denjenigen Clipping-Ebenen berechnet werden, die zwischen den Endpunkten der Strecke liegen (was wegen der vorher erfolgten Normierung des Sichtquaders sehr einfach ist). Anhand der Koordinaten der Schnittpunkte kann entschieden werden, ob eine (und wenn ja, welche) durch die Schnittpunkte entstandene Teilstrecke innerhalb des Würfels liegt.

Beim Clipping von Polygonen wird für eine der Clipping-Ebenen das „vor“ der Ebene liegende Teilpolygon bestimmt, dieses an einer weiteren Ebene geclippt und – nach Wiederholung des Vorgangs für alle 6 Clipping-Ebenen – somit das innerhalb des Sichtvolumens befindliche Teilpolygon ermittelt.<sup>84</sup>

<sup>83</sup>Um diese Entscheidungen schnell zu treffen, wird bei dem meist verwendeten COHEN-SUTHERLAND-ALGORITHMUS jedem Endpunkt ein aus 6 Bit bestehender Code zugeordnet, der angibt, vor bzw. hinter welcher der Clipping-Ebenen der betreffende Punkt liegt, siehe z. B. [57], S. 282, 288.

<sup>84</sup>Hierfür kommt oft der z. B. in [11], S. 49 und [57], S. 286 beschriebene SUTHERLAND-HODGMAN-Algorithmus zur Anwendung.

Es mag auf den ersten Blick sinnvoll erscheinen, zunächst die Projektion der Szene in eine Ebene durchzuführen und danach das Clipping am Sichtfenster, also einem Rechteck, vorzunehmen. Natürlich sind die dabei nötigen Berechnungen schneller auszuführen und es entfallen bei Zentralprojektionen die Transformation des Sichtvolumens in einen Quader und die inverse Transformation (welche nach dem Clipping an dem Quader erforderlich ist). Allerdings gehen bei der Projektion die Tiefeninformationen verloren, die für die Entscheidung notwendig sind, welche Objekte von anderen Objekten verdeckt werden. Es ist i. Allg. effizienter, zunächst das Clipping vorzunehmen und erst danach verdeckte Objekte zu entfernen, weshalb „Computergrafik-Pipelines“ meist in dieser Reihenfolge konzipiert werden (siehe z. B. [11], S. 72).

### 2.4.5 Entfernung verdeckter Objekte und Facetten

Eine wichtige Bedeutung für die Bildberechnung bei „schnellen“ 3D-Grafikanwendungen haben Algorithmen, welche die Sichtbarkeit von Objekten ermitteln und der Kamera abgewandte Facetten sowie verdeckte Objekte entfernen.<sup>85</sup> Im Folgenden wird ein Überblick über einige der vielfältigen hierfür angewendeten Verfahren<sup>86</sup> gegeben, wobei als Objekte lediglich Polygone betrachtet werden, da in der Praxis bei schnellen Anwendungen die Repräsentation von Objekten durch Polygonnetze zum Einsatz kommt.

Bei konvexen Polyedern ist in der Regel etwa die Hälfte der Begrenzungsflächen<sup>87</sup> für den Betrachter bzw. die Kamera unsichtbar. Wenn konsistent orientierte Dreiecke<sup>88</sup> vorliegen, lassen sich sichtbare Facetten durch die Berechnung der Skalarprodukte nach außen gerichteter Normalenvektoren  $\vec{n}$  mit Richtungsvektoren  $\vec{s}$  von Sehstrahlen, die durch Punkte der Dreiecke verlaufen, bestimmen. Ist  $\langle \vec{n}, \vec{s} \rangle < 0$ , so ist das entsprechende Dreieck sichtbar, für  $\langle \vec{n}, \vec{s} \rangle = 0$  ist es lediglich als Kante sichtbar und für  $\langle \vec{n}, \vec{s} \rangle > 0$  verdeckt. In dem in Abb. 2.54 a) dargestellten Beispiel ist  $\langle \vec{n}_1, \vec{s}_1 \rangle < 0$ ,  $\langle \vec{n}_4, \vec{s}_4 \rangle < 0$  sowie  $\langle \vec{n}_2, \vec{s}_2 \rangle > 0$  und  $\langle \vec{n}_3, \vec{s}_3 \rangle > 0$ ; die durch gestrichelte Strecken symbolisierten Dreiecke sind demnach verdeckt und können aus der Objektliste entfernt werden.

<sup>85</sup>In der fotorealistischen Computergrafik dürfen allerdings verdeckte Objekte bzw. Facetten aus den bereits in Fußnote 81 auf S. 119 genannten Gründen nicht entfernt werden. Die Sichtbarkeit wird bei dem für diesen Anwendungsbereich hauptsächlich verwendeten Raytracing-Verfahren direkt durch die zur Bildberechnung durchgeführte Strahlverfolgung ermittelt (wie in Abschnitt 2.5.5 deutlich wird).

<sup>86</sup>Siehe u. a. [79], S. 499–538; [195], S. 229–242; [248], S. 104–114 sowie [282], S. 249–265.

<sup>87</sup>Bevorzugt werden die begrenzenden Polygone in Dreiecke unterteilt, siehe Abschnitt 2.3.9.

<sup>88</sup>Für diese Dreiecke sind leicht Normalenvektoren zu ermitteln, die „nach außen“ zeigen, vgl. S. 110.

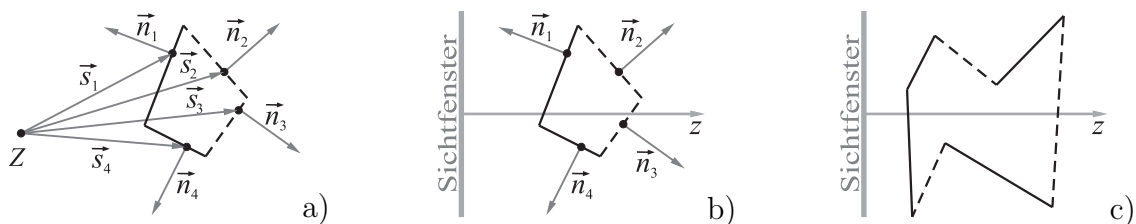


Abbildung 2.54: Entfernung von Rückseiten (Backface Culling)

Besonders einfach ist die Entfernung von Rückseiten (meist als *Backface Culling* bezeichnet) bei einer Parallelprojektion oder nach Transformation der Szene in ein normiertes Sichtvolumen entsprechend Abb. 2.53 auf S. 120. In diesem Falle ist ein Dreieck genau dann unsichtbar, falls sein (nach außen gerichteter) Normaleneinheitsvektor eine positive  $z$ -Komponente besitzt (siehe Abb. 2.54 b). Für nicht konvexe Polyeder kann es bei diesem einfachen Verfahren jedoch auftreten, dass verdeckte Polygone dennoch nicht entfernt werden (siehe Abb. 2.54 c). Diese sind dann in den folgenden Sichtbarkeitsuntersuchungen weiter zu berücksichtigen. Eine vollständige Aussage über sichtbare Objekte ist nach Durchführung des Backface Culling lediglich für Szenen möglich, die nur aus einem einzigen konvexen Polyeder bestehen. Für komplexere Szenen verkürzt dieses einfache Verfahren jedoch erheblich die Zeit für die weiteren Sichtbarkeitsuntersuchungen, da durch die Entfernung der Rückseiten i. Allg. wesentlich weniger Polygone in die Berechnungen einzubeziehen sind.

Die bisher beschriebenen Verfahren zur Sichtbarkeitsbestimmung (Clipping, Backface Culling) werden anhand der geometrischen Objekte, aus denen eine Szene besteht, ausgeführt (bzw. anhand ihrer Facetten, die wiederum geometrische Objekte, meist Dreiecke, sind). Diese Verfahren werden deshalb als *Objektraum-Algorithmen* bezeichnet. Dazu gehören auch Algorithmen, welche die Objekte (Dreiecke) einer Szene nach ihren Tiefenwerten ( $z$ -Koordinaten) sortieren und dann – beginnend bei den am weitesten hinten liegenden – darstellen, wobei weiter vorn liegende Dreiecke die dahinter liegenden überdecken.<sup>89</sup> Dies führt jedoch nur dann zu einer fehlerfreien Darstellung, wenn keine Facetten existieren, die andere Facetten schneiden, diese also in den Tiefenwerten (und auch hinsichtlich der anderen Koordinatenintervalle, in welchen die Punkte der Facetten liegen) überlappen. Tritt dieser Fall auf, so müssen durch Schnittberechnungen die von den beteiligten Polygonen sichtbaren Teilpolygone ermittelt werden, was zu einem recht hohen Aufwand führt.

Ein Beispiel für die Anwendung des Painter-Algorithmus ist der Grafikexport des CAS MuPAD. Exportiert man in diesem Programm eine dreidimensionale Grafik als Vektordatei (in einem der Formate eps, emf oder wmf), öffnet diese in einem Vektorgrafikprogramm und entfernt Polygone, so kommen die dahinter liegenden, verdeckten Polygone zum Vorschein. Fehler an Schnittkurven von Flächen sind deutlich erkennbar. Dies betrifft jedoch nur exportierte Grafiken; die Grafikdarstellung in MuPAD selbst erfolgt unter Verwendung der OpenGL-Bibliotheken (siehe S. 161), wobei der im Folgenden beschriebene  $z$ -Buffer-Algorithmus zur Sichtbarkeitsbestimmung Anwendung findet.

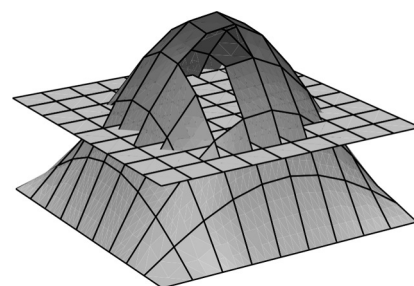


Abbildung 2.55: Als Vektordatei exportierte MuPAD-Grafik, aus der in Corel Draw Polygone entfernt wurden

<sup>89</sup>Diese Vorgehensweise ist der Landschaftsmalerei alter Meister ähnlich, die ebenfalls zunächst weit entfernte Landschaftsteile darstellten und diese dann mit Vordergrundobjekten übermalten, daher werden derartige Tiefensortier-Algorithmen auch als Maler- bzw. *Painter-Algorithmen* bezeichnet.



Von einem völlig anderen Ansatz als Objektraum-Algorithmen gehen pixelorientierte Verfahren (oft als *Bildraum-Algorithmen* bezeichnet) aus. Hierbei wird für jedes Pixel des zu berechnenden Bildes (also für eine diskrete Menge von Punkten des Sichtfensters, siehe Abb. 2.56) ermittelt, für welches Objekt der Schnittpunkt mit dem durch das betrachtete Pixel verlaufenden Projektionsstrahl dem Sichtfenster am nächsten liegt. Derartige Methoden beziehen sich somit auf die gewünschte Auflösung des Bildes (Anzahl der Pixelzeilen und -spalten), die sich häufig nach der Auflösung des Bildschirmes richtet, und werden deshalb als *geräteabhängig* bezeichnet.

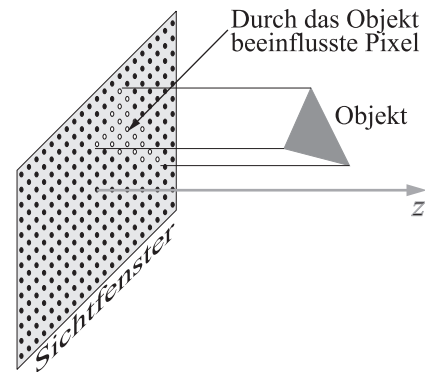


Abbildung 2.56: Sichtfenster mit Pixelraster

Eine besonders große Verbreitung unter den Bildraum-Algorithmen hat der *z-Buffer-Algorithmus* erlangt. Zunächst sind dabei alle Objekte (Polygone) der Szene, welche nicht bereits durch Clipping oder Backface Culling entfernt wurden, auf das Sichtfenster zu projizieren und die durch die jeweiligen Polygone überdeckten Pixel zu ermitteln.<sup>90</sup> Der *z-Buffer-Algorithmus* besteht dann aus folgenden Schritten:

- Anlegen einer Matrix  $Z^{buf} = (z_{x,y}^{buf})$ , deren Zeilen- und Spaltenanzahlen den Anzahlen der Pixelzeilen bzw. -spalten des Bildes entsprechen, als Zwischenspeicher für die Tiefenwerte (*z-Buffer*) und Eintragen der *z*-Koordinate  $z_{back}$  der hinteren Clipping-Ebene für alle Matrixelemente;<sup>91</sup>
- Zuordnung einer Hintergrundfarbe zu allen Pixeln des Bildes;
- für jedes projizierte Polygon:
  - für jedes Pixel  $(x, y)$ , welches durch das Polygon überdeckt wird:
    - Bestimmung der Tiefe  $z(x, y)$  des Polygons im Pixel  $(x, y)$ ;
    - falls  $z(x, y) < z_{x,y}^{buf}$ : neue Belegung des Matrixelements  $z_{x,y}^{buf}$  mit  $z(x, y)$  und Zuordnung der Farbe des Polygonpunktes zu dem Pixel  $(x, y)$ ;
    - falls  $z(x, y) \geq z_{x,y}^{buf}$ : keine Änderungen an  $Z^{buf}$  und an der Farbe.

Nach Durchführung dieser Schritte für alle Polygone hat jedes Pixel die Farbe des aus der Sicht der Kamera nächsten Objekts. Die Reihenfolge, in der die Polygone durchlaufen werden, ist (außer bei Polygonpunkten mit gleichen Tiefenwerten) beliebig.

Der *z-Buffer-Algorithmus* und Varianten davon haben sich für Echtzeit-3D-Darstellungen (wie bei Spielen und zur Vorschau in 3D-Grafiksoftware sowie CAD-Systemen) als dominierende Sichtbarkeitsverfahren durchgesetzt und werden hauptsächlich durch sehr schnelle, darauf spezialisierte Bausteine durchgeführt, die seit Ende der neunziger Jahre auch auf den Grafikkarten üblicher Personalcomputer vorhanden sind.

<sup>90</sup>Dabei wird für jedes Polygon und jede Pixelzeile durch Bestimmung der Schnittpunkte der Kanten des projizierten Polygons mit der Pixelzeile ermittelt, welche Pixel innerhalb des projizierten Polygons liegen. Dieser Vorgang wird als Scan-Konvertierung bezeichnet, siehe z. B. [248], S. 106.

<sup>91</sup>Hinter der hinteren Clipping-Ebene liegende Punkte (mit  $z > z_{back}$ ) sind nicht sichtbar, siehe 2.4.4.

### 2.4.6 Texture-Mapping (Projektion von Bildern auf Flächen)

Um Körper bzw. Flächen mit Oberflächenerscheinungen zu versehen, werden oft Fotos oder andere Bilder auf die Objekte projiziert. Dieses erstmals 1974 von CATMULL vorgeschlagene Verfahren ist als essenziell für die Herausbildung der fotorealistischen 3D-Computergrafik anzusehen. Die projizierten Bilder werden als *Texture Maps*, oft auch als *Texturen*, bezeichnet – wobei letzterer Begriff allgemeiner ist und auch synthetische, durch Prozeduren generierte Farbanordnungen und Strukturen umfasst. Bei einer Texturprojektion erfolgt eine Abbildung der Punkte eines Rechtecks (des zweidimensionalen Pixelbildes) auf die Punkte einer Fläche im Raum. Bei dieser Abbildung werden den Punkten der Fläche die Helligkeits- bzw. Farbwerte der Bilddatei zugeordnet.

Im einfachsten Falle, bei der Projektion eines Bildes auf eine ebene Oberfläche, ist eine affine Abbildung aus der zweidimensionalen Bildebene in den Raum vorzunehmen. So bildet z. B. POV-Ray bei *ebenen Texturprojektionen* Bilder generell auf das Quadrat mit den Eckpunktkoordinaten  $(0; 0; 0)$ ,  $(0; 1; 0)$ ,  $(1; 0; 0)$  und  $(1; 1; 0)$  ab. Wird die Bilddatei bei der Projektion skaliert, so erscheint sie mehrfach bzw. nur teilweise in diesem Quadrat. Durch weitere Transformationen kann das Objekt mitsamt der Textur verschoben, gedreht und skaliert werden, ohne dass sich etwas an der Zuordnung des Bildes zu dem Objekt ändert. In Abb. 2.57 a) sind drei sehr dünne Quader dargestellt, auf die das Logo der Humboldt-Universität mit den Streckungsfaktoren 1 (oben),  $\frac{1}{2}$  (rechts) sowie  $\frac{1}{3}$  (vorn) projiziert wurde. Da die Projektion immer in die  $x$ - $y$ -Ebene erfolgt, mussten zwei der Quader nach der Projektion des Logos gedreht werden. So wurde die rechte Seitenfläche durch folgende Eingaben erzeugt:

```
box{ <0,0,-0.0001> <1,1,0.0001>
    texture{pigment{image_map tiff "HU-Logo.tif" map_type 0 scale 1/2} }
    rotate <0,-90,0> translate <1,0,0> }92
```

<sup>92</sup>Die Skalierung innerhalb der **texture**-Anweisung wirkt sich lediglich auf die Textur aus, während sich die Drehung und die Verschiebung außerhalb des **texture**-Blocks auf Objekt und Textur beziehen.

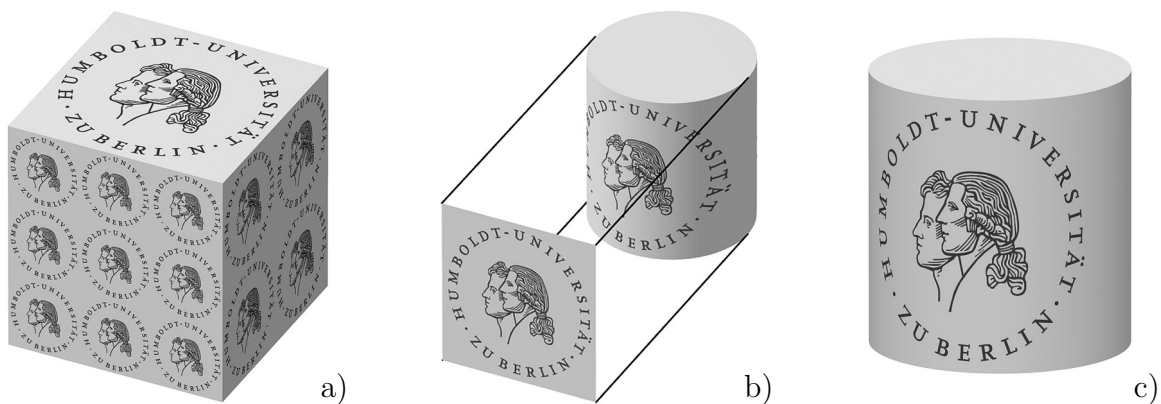


Abbildung 2.57: ebene (a, b) sowie zylindrische Texturprojektion (c)

Farbgrafiken: siehe S. 352; Dateien: **Texturprojektion1-3.pov** (siehe Anhang D, S. 365)

Mittels ebener Texturprojektionen (in POV-Ray durch `map_type 0` gekennzeichnet) können Bilder auf beliebige Objekte projiziert werden; allerdings erfolgt dabei eine senkrechte Parallelprojektion, was zu Verzerrungen der Bilder auf gekrümmten Oberflächen führt, wie Abb. 2.57 b) für einen Zylinder zeigt, auf den ein Bild auf diese Weise projiziert wurde. Mittels einer *Zylinderprojektion* ist es möglich, Bilder verzerrungsfrei (d. h. längenverhältnistreue) auf Zylinder abzubilden. Damit ein Bild verzerrungsfrei um einen Zylinder „gewickelt“ werden kann, müssen seine Abmessungen im Verhältnis  $2\pi r : h$  zueinander stehen (wobei  $r$  der Radius des Zylinders und  $h$  seine Höhe ist). Die Projektion der Punkte  $P(u; v)$  eines solchen Bildes (mit  $u \in [0; s \cdot 2\pi r]$ ,  $v \in [0; s \cdot h]$ ) auf einen Zylinder mit den Mittelpunkten  $M_1(0, 0, 0)$  und  $M_2(0, h, 0)$  der Grund- bzw. Deckfläche sowie dem Radius  $r$  erfolgt dann durch die Abbildungsgleichungen

$$x(u, v) = r \cdot \cos \frac{u}{s \cdot r}, \quad y(u, v) = \frac{v}{s} \quad \text{und} \quad z(u, v) = r \cdot \sin \frac{u}{s \cdot r}.$$

Abb. 2.57 c) wurde mithilfe einer derartigen zylindrischen Projektion in POV-Ray erzeugt (dazu ist `map_type 2` zu setzen). Die Positionierung und Skalierung des Zylinders (mit projiziertem Bild) kann wiederum durch affine Transformationen erfolgen.<sup>93</sup>

Grundsätzlich unmöglich ist es, ebene Bilder verzerrungsfrei auf Kugeloberflächen abzubilden.<sup>94</sup> Dennoch werden häufig Bilder mittels *sphärischer Projektionen* auf Kugeln projiziert, wobei eine Zuordnung der ebenen Koordinaten  $u$  und  $v$  eines Pixelbildes zu den Koordinaten  $\lambda$  und  $\varphi$  eines Kugelkoordinatensystems (siehe Abb. 2.58) erfolgt:

$$\begin{aligned} x(u, v) &= r \cdot \cos \lambda \cdot \cos \varphi = r \cdot \cos \left( \frac{u - \frac{b}{2}}{b} \cdot 2\pi \right) \cdot \cos \left( \frac{v - \frac{h}{2}}{h} \cdot \pi \right), \\ z(u, v) &= r \cdot \sin \lambda \cdot \cos \varphi = r \cdot \sin \left( \frac{u - \frac{b}{2}}{b} \cdot 2\pi \right) \cdot \cos \left( \frac{v - \frac{h}{2}}{h} \cdot \pi \right), \\ y(u, v) &= r \cdot \sin \varphi = r \cdot \sin \left( \frac{v - \frac{h}{2}}{h} \cdot \pi \right). \end{aligned}$$

Dabei sind  $b$  und  $h$  die Breite und die Höhe des projizierten Bildes; es ist also  $u \in [0; b]$ ,  $v \in [0; h]$ . Die Kugel wird damit vollständig durch das Bild „bedeckt“, denn die Definitionsbereiche der Kugelkoordinaten  $\lambda$  und  $\varphi$  sind  $[-\pi; \pi)$  bzw.  $[-\pi/2; \pi/2]$ .

Damit in Äquatornähe eine annähernd längenverhältnistreue Abbildung erreicht wird, muss  $b : h = 2 : 1$  gelten. Abb. 2.59 a) zeigt die sphärische Projektion (in POV-Ray `map_type 1`) eines gleichmäßigen Gitternetzes, das diese Bedingung erfüllt, auf eine Kugel. Die Linien des Gitternetzes (für die jeweils  $u$  oder  $v$  konstant ist) werden dabei

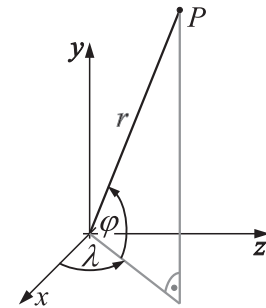


Abbildung 2.58:  
Kugelkoordinaten

<sup>93</sup>Natürlich wird das Bild durch die Projektion des Zylinders in eine Ebene letztendlich verzerrt abgebildet. Die Abbildung des Bildrechtecks auf den Zylinder erfolgte jedoch längenverhältnistreue.

<sup>94</sup>Nach dem Theorema Egregium von GAUSS bleibt die Krümmung von Flächen bei längentreuen Flächenabbildungen invariant. Eine längenverhältnistreue Abbildung von einer Fläche der Krümmung Null (wie einer Ebene oder auch einer Zylinderfläche) auf eine Sphäre, bei der es sich um eine Fläche konstanter positiver Gaußscher Krümmung handelt, ist somit nicht möglich.

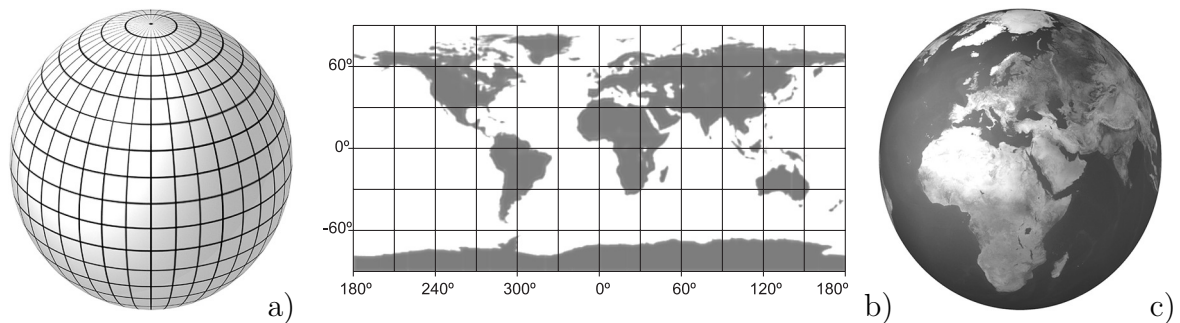


Abbildung 2.59: Sphärische Projektionen, quadratische Plattkarte

Farbgrafiken: siehe S. 352; Dateien: `Texturprojektion4-5.pov`

auf die Meridiane und Breitenkreise abgebildet, wobei die Abbildung der Koordinatenlinien mit  $u = \text{const}$  auf die Meridiane längen(verhältnis-)treu erfolgt. Steht eine Abbildung (Karte) der Erde zur Verfügung, bei der die Längen- und Breitenkreise auf ein Netz quadratischer Maschen abgebildet wurden (Abb. 2.59 b),<sup>95</sup> so ergibt eine Projektion auf eine Kugel mit den auf S. 125 angegebenen Projektionsgleichungen eine Ähnlichkeitsabbildung der Erde auf diese Kugel, wie in Abb. 2.59 c) dargestellt ist.<sup>96</sup>

Die allgemeinste Art der Texturprojektion ist das so genannte *u-v-Mapping*, bei dem eine Zuordnung der Koordinaten eines zweidimensionalen Bildes zu den Parametern der Fläche erfolgt, auf die das Bild projiziert wird.<sup>97</sup> In den meisten 3D-Grafikprogrammen ist die Zuordnung von Texturen zu den *u-v*-Koordinaten nur für Polygonnetze möglich;<sup>98</sup> vor der Anwendung von *u-v*-Mapping-Techniken ist also meist eine Tessellation von Flächen vorzunehmen (siehe 2.3.9). Punkte der Bilddatei werden dann Parameterpaaren  $(u; v)$  und diese den Eckpunkten eines Polygonnetzes zugeordnet.<sup>99</sup> Zwischen den Eckpunkten werden die Pixel der aufzubringenden Bilddatei, die sich zwischen denjenigen Pixeln befinden, denen die  $(u; v)$ -Paare der Eckpunkte entsprechen, interpoliert.<sup>100</sup>

Eine mögliche Vorgehensweise beim *u-v*-Mapping sei anhand des Aufbringens des Logos der Humboldt-Universität zu Berlin auf die Teekanne der Universität Utah<sup>101</sup> beschrieben, welches mithilfe der 3D-Grafiksoftware Carrara erfolgte. Der Bauch der Teekanne wurde zunächst in ein Polygonnetz mit relativ wenigen Eckpunkten (siehe Abb. 2.60

<sup>95</sup>Eine derartige Karte der Erde wird als *quadratische Plattkarte* bezeichnet, siehe [222], S. 76–79.

<sup>96</sup>Unebenheiten und die nicht exakte Kugelform der realen Erde werden dabei nicht berücksichtigt.

<sup>97</sup>Siehe dazu auch Fußnote 43 auf S. 94.

<sup>98</sup>POV-Ray unterstützt *u-v*-Mapping auch für einige weitere Objekte (siehe [200], S. 191); die Handhabung ist jedoch wenig praktikabel, es wird eher die prinzipielle Möglichkeit demonstriert.

<sup>99</sup>In den gebräuchlichen *u-v*-Mapping-Werkzeugen ist es möglich, Polygone zu unterteilen, um eine höhere Genauigkeit der Zuordnung zu erreichen.

<sup>100</sup>Für qualitativ hochwertige Texturprojektionen sind (auch bei sphärischen und zylindrischen Projektionen) aufwändige Interpolations- und Filteralgorithmen erforderlich, siehe u. a. [189], S. 211–274. Derartige Algorithmen werden inzwischen zunehmend in den Bausteinen von Grafikkarten implementiert, um auch für Echtzeit-3D-Anwendungen eine hochwertige Darstellungsqualität zu ermöglichen.

<sup>101</sup>Siehe die Bemerkung auf S. 107 und Abb. B.14 auf S. 351 in Anhang B.

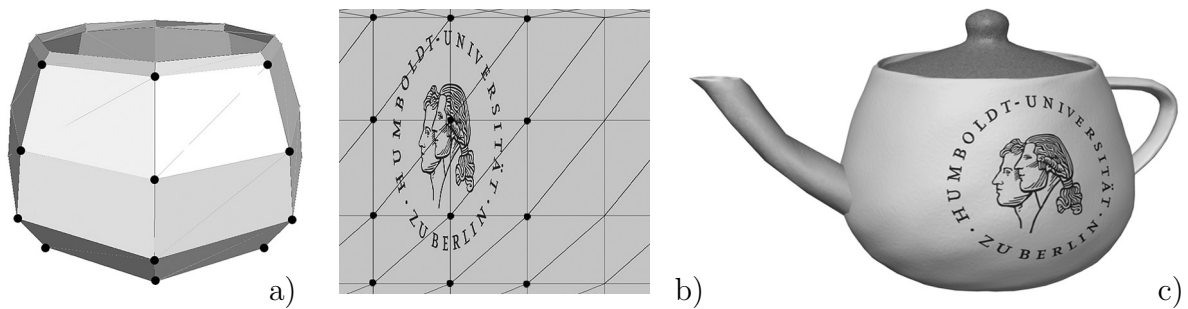


Abbildung 2.60: UV-Mapping

a) umgewandelt, um besser die Übersicht zu behalten. Der „*u-v*-Editor“ von Carrara kann aus einem solchen Netz eine ebene Darstellung der *u-v*-Koordinaten (Parameter) der Eckpunkte erzeugen – wie in Abb. 2.60 b) dargestellt, jedoch noch ohne Logo.<sup>102</sup> Diese Darstellung wurde exportiert und in einem Bildbearbeitungsprogramm das Logo an die gewünschte Stelle montiert. Um eine annähernd unverzerrte Abbildung im letztendlichen Bild<sup>103</sup> zu erhalten, musste das Logo dabei verzerrt werden, wie in Abb. 2.60 b) sichtbar ist. Nachdem sich das Logo an der vorgesehenen Position befand und nach dem Ausblenden der Parameterlinien und Punkte wurde die so entstandene Bilddatei in Carrara als Texture Map geladen und schließlich das Polygonnetz mithilfe des Subdivision-Verfahrens (siehe S. 111) geglättet, damit es wieder annähernd die Gestalt der ursprünglichen Teekanne annimmt. Das nach dieser Folge von Arbeitsschritten berechnete Bild der Teekanne zeigt Abb. 2.60 c) bzw. (besser erkennbar) die entsprechende farbige Abb. auf S. 352 in Anhang B.<sup>104</sup>

<sup>102</sup>Die Zuordnung der Ecken des Netzes zu den Schnittpunkten der Parameterlinien wird durch Markieren („Anklicken“) sichtbar; einige einander entsprechende Eckpunkte und die zugehörigen „Parameterpunkte“ sind in Abb. 2.60 links bzw. Mitte markiert.

<sup>103</sup>Ein völlig unverzerrte, längenverhältnistreue Projektion eines ebenen Bildes auf eine gekrümmte Fläche (d. h. eine Fläche, die nicht überall die Gaußsche Krümmung Null hat) ist unmöglich, siehe Fußnote 94 auf S. 125. Daher kann die Projektion des Logos auf die Teekanne (die nach der abschließend erfolgten Glättung wieder als gekrümmte Fläche aufzufassen ist) nicht verzerrungsfrei sein.

<sup>104</sup>Das geschilderte Beispiel deckt nur recht elementare Vorgehensweisen bei der Texturerstellung mittels *u-v*-Mapping ab. So kann z. B. durch Manipulation (Ziehen) von Eckpunkten in dem verwendeten *u-v*-Editor von Carrara die Zuordnung der Bildbereiche zu den Eckpunkten verändert werden. Eine etwas ausführlichere Beschreibung dieses Werkzeugs anhand der Texturierung menschlicher Gesichter enthält [45], S. 272–279. Einige Programme (wie Maxon Bodypaint 3D, siehe [www.maxon.net](http://www.maxon.net)) ermöglichen sogar die interaktive, Polygonnetze und Texturen wechselseitig beeinflussende, Anwendung von Bildbearbeitungs- und Malwerkzeugen. Dennoch erfordert die Texturerstellung durch *u-v*-Mapping einen sehr hohen Einarbeitungs- und Arbeitsaufwand, wenn hochwertige, realistische Ergebnisse erzielt werden sollen. Bei professionellen Produktionen arbeiten mehrköpfige Teams von Spezialisten an dieser Aufgabe.

## 2.5 Erzeugung fotorealistischer 3D-Darstellungen

Durch die in den vorangegangenen Abschnitten beschriebenen Projektionen und Sichtbarkeitsuntersuchungen können Objekte in eine Bildebene abgebildet und dabei Farbwerte von Punkten der Objekte den entsprechenden Punkten des Bildes zugeordnet werden. (Die Farben der Punkte von Objekten ergeben sich aus den festgelegten Objektfarben oder aus Farben der Pixel von Bildern, die durch Texturprojektion aufgebracht wurden.) Allerdings entsteht dabei kein räumlicher Eindruck, das Bild einer einfarbigen Kugel ist z. B. eine Kreisscheibe mit konstanter Farbe (siehe Abb. 2.61).

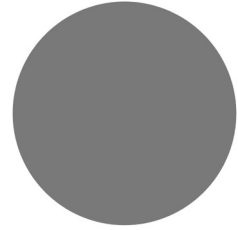


Abbildung 2.61:  
Bild einer Kugel

Die Wahrnehmung von Körpern bzw. Flächen in der Realität hängt nicht nur von deren Farben, sondern entscheidend auch davon ab, wie sie beleuchtet werden, d. h. wo sich Lichtquellen befinden, mit welcher Intensität und Farbe sie Licht ausstrahlen und wie die Oberflächen der Objekte die Lichtstrahlen reflektieren, absorbieren oder transmittieren. Die Simulation von Beleuchtung zählt damit zu den wichtigsten Aufgaben der fotorealistischen Computergrafik. Eine Lichtquelle wird in der Computergrafik meist als punktförmig angenommen und durch ihre Position sowie die Farbe des emittierten Lichtes beschrieben. Die Beschreibung einer Lichtquelle in POV-Ray erfolgt durch

```
light_source { <x,y,z> color rgb <r,g,b> };
```

dabei geben  $x,y,z$  die Koordinaten der Lichtquelle sowie  $r,g,b$  ihre Farbintensitäten im RGB-Modell (siehe Abschnitt 2.1.2) an.

### 2.5.1 Reflexion, Transparenz und Brechung

Der Bildberechnung in der fotorealistischen Computergrafik liegen die physikalischen Gesetze der Lichtausbreitung zugrunde, von denen die elementarsten das Reflexions- und das Brechungsgesetz sind.

Das *Reflexionsgesetz* besagt, dass der einfallende und der reflektierte Lichtstrahl in einem Punkt einer Oberfläche sowie die Senkrechte auf der Fläche in diesem Punkt (als Einfallslot bezeichnet) in einer Ebene liegen und die Winkel zwischen dem Einfallslot und dem einfallenden Lichtstrahl sowie zwischen dem Einfallslot und dem reflektierten Lichtstrahl gleich groß sind.

Vektoriell kann das Reflexionsgesetz folgendermaßen ausgedrückt werden: Ist  $\vec{n}$  Normaleneinheitsvektor einer spiegelnden Oberfläche und sind  $\vec{l}$  und  $\vec{b}$  die normierten Richtungsvektoren des einfallenden bzw. des reflektierten Lichtstrahls, so gilt:

- $\vec{n}$ ,  $\vec{l}$  und  $\vec{b}$  sind komplanar,
- $\langle \vec{l}, \vec{n} \rangle = \langle \vec{b}, \vec{n} \rangle$ .

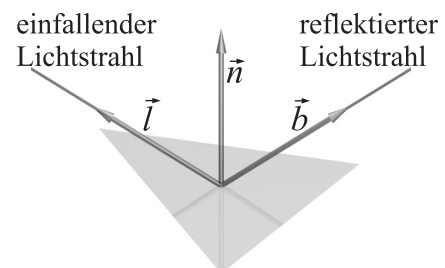


Abbildung 2.62: Reflexionsgesetz



Die Intensität  $I$ , mit der in einem Punkt Licht einer Lichtquelle (die auch ein Licht reflektierender Punkt einer anderen oder derselben Objektoberfläche sein kann) zu einem Beobachter reflektiert wird, kann somit durch

$$I = \begin{cases} I_{\text{ein}} \cdot k_s & \text{falls } \vec{n} = \frac{\vec{b} + \vec{l}}{|\vec{b} + \vec{l}|} \\ 0 & \text{sonst} \end{cases} \quad \text{mit } 0 \leq k_s \leq 1. \quad (2.15)$$

angegeben werden, wobei  $\vec{l}$  und  $\vec{b}$  die normierten Verbindungsvektoren des betrachteten Punktes mit der Lichtquelle bzw. der Kamera sind. Die Konstante  $k_s$  gibt an, wie stark die Oberfläche eines Materials Lichtstrahlen direkt reflektiert, für ideale Spiegel gilt  $k_s = 1$ . In POV-Ray kann diese Konstante durch die Anweisung `reflection` für jedes Objekt festgelegt werden. Abb. 2.63 zeigt eine mithilfe von POV-Ray generierte spiegelnde Kugel mit  $k_s = 0.3$ , wobei die Kugel nicht nur durch direkte Reflexion, sondern auch durch die im folgenden Abschnitt beschriebene diffuse Komponente beleuchtet wird.

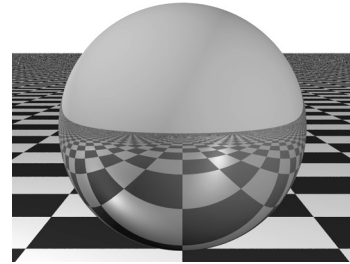


Abbildung 2.63: Reflexion

Farbige Versionen dieser Abb. sowie aller anderen in diesem Abschnitt abgebildeten Beispiele befinden sich auf S. 353 in Anhang B.

Aufgrund der Tatsache, dass sich die Geschwindigkeit des Lichtes beim Übergang von einem Medium in ein anderes ändert, erfolgt an der Grenzfläche eine Richtungsänderung von Lichtstrahlen, die als *Refraktion* oder *Brechung* bezeichnet wird. Lichtbrechung hat somit in der Computergrafik nur bei der Betrachtung (teil-)transparenter Objekte eine Bedeutung. Der *Brechungsindex*  $n$  eines durchsichtigen Materials ist der Quotient der Lichtgeschwindigkeit im Vakuum und der in dem betrachteten Material.<sup>105</sup> Das *Brechungsgesetz* beinhaltet zwei Aussagen:

- Einfallender und gebrochener Lichtstrahl in einem Oberflächenpunkt sowie das Einfallslot in diesem Punkt liegen in einer Ebene.
- Für die Winkel  $\alpha_1$  und  $\alpha_2$  zwischen dem Einfallslot und dem einfallenden bzw. dem gebrochenen Lichtstrahl sowie die Brechungsindizes  $n_1$  des Materials, aus dem der Strahl kommt, und  $n_2$  des Materials, in das er übergeht, gilt

$$\frac{\sin \alpha_1}{\sin \alpha_2} = \frac{n_2}{n_1}. \quad (2.16)$$

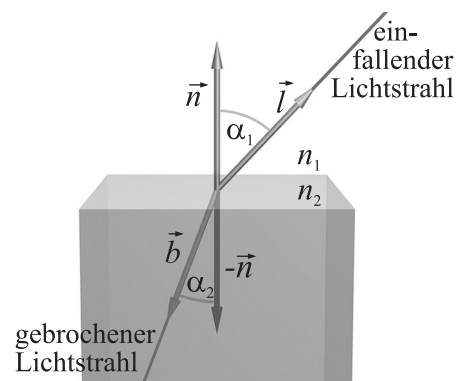


Abbildung 2.64: Brechungsgesetz

Vektoriell kann das Brechungsgesetz in der Form

$$n_1 \cdot (\vec{n} \times \vec{l}) = n_2 \cdot (-\vec{n} \times \vec{b}) \quad (2.17)$$

<sup>105</sup>Luft hat einen Brechungsindex von ca. 1,00029; der Lichtgeschwindigkeitsunterschied zwischen Luft und Vakuum ist also sehr gering. Demgegenüber beträgt der Brechungsindex von Wasser ca. 1,33, womit sich Licht in Wasser nur etwa 3/4 so schnell ausbreitet wie im Vakuum oder in der Luft.

geschrieben werden, wobei  $\vec{n}$  der Normaleneinheitsvektor der Grenzfläche im Auftreffpunkt des einfallenden Lichtstrahls ist sowie  $\vec{l}$  und  $\vec{b}$  die normierten Richtungsvektoren des einfallenden bzw. des gebrochenen Lichtstrahls sind. Da das Vektorprodukt zweier Vektoren zu beiden der Vektoren, aus denen es gebildet wird, orthogonal ist, beinhaltet Gleichung (2.17) auch die Bedingung, dass die Vektoren  $\vec{n}$ ,  $\vec{l}$  und  $\vec{b}$  komplanar sind.

Die Simulation „überzeugender“ Transparenz von Körpern in computergrafischen Darstellungen setzt die Berücksichtigung des Brechungsgesetzes voraus. Es ist leicht möglich, Objekte durchsichtig erscheinen zu lassen; in POV-Ray erfolgt dies z. B. dadurch, dass der Beschreibung der Objektfarbe durch die RGB-Komponenten (siehe 2.1.2) eine vierte Komponente  $f$  bzw.  $t$  (für gefilterte bzw. transmittierende Transparenz<sup>106</sup>) hinzugefügt wird. Für die Erzeugung von Abb. 2.65 a) wurde lediglich die Farbbeschreibung der Kugel um den Transparenzwert  $t = 0,4$  ergänzt. Um auf realistische Weise Glaskörper oder Wasser nachzubilden, ist es jedoch zusätzlich notwendig, die Brechung zu simulieren. In POV-Ray geschieht dies durch Einfügen der Anweisung

`interior { ior Brechungsindex }`

in die Objektbeschreibung.<sup>107</sup> Die Abbildungen 2.65 b) und c) wurden mit dem Wert `ior = 1,5` generiert, der dem Brechungsindex von Glas entspricht.

Die physikalisch korrekte Simulation von Reflexion und Brechung verursacht einen hohen Rechenaufwand und bleibt daher (zumindest mittelfristig) fotorealistischer, meist nach dem Raytracing-Verfahren arbeitender Grafiksoftware vorbehalten, die zum Teil viel Zeit für die Bildberechnung benötigt. Dennoch scheinen oft auch in Computerspielen, die Bilder in Echtzeit berechnen, Reflexionen und sogar Brechungen aufzutreten. Hierbei handelt es sich jedoch nicht um physikalisch exakt simulierte Effekte. Vielmehr nutzen Spieleprogrammierer vorgefertigte Bilder (Reflection- bzw. Environment-sowie Refraction-Maps) und projizieren diese als Texturen auf die Objekte (siehe 2.4.6). Damit lassen sich auch Reflexionen von nicht existierenden Objekten vortäuschen.

<sup>106</sup>Bei der gefilterten Transparenz wird die Farbe dahinter liegender Objekte durch die Farbe des durchdrungenen Objekts gefiltert, was u. a. bei farbigem Glas der Fall ist. Transmittierende Transparenz bewirkt keine Farbfilterung; diese Transparenzart tritt z. B. bei sehr dünnen Textilien auf.

<sup>107</sup>Die Bezeichnung `interior` gibt an, dass es sich nicht um eine Oberflächen-, sondern um eine Materialeigenschaft handelt; `ior` steht für Index Of Refraction.

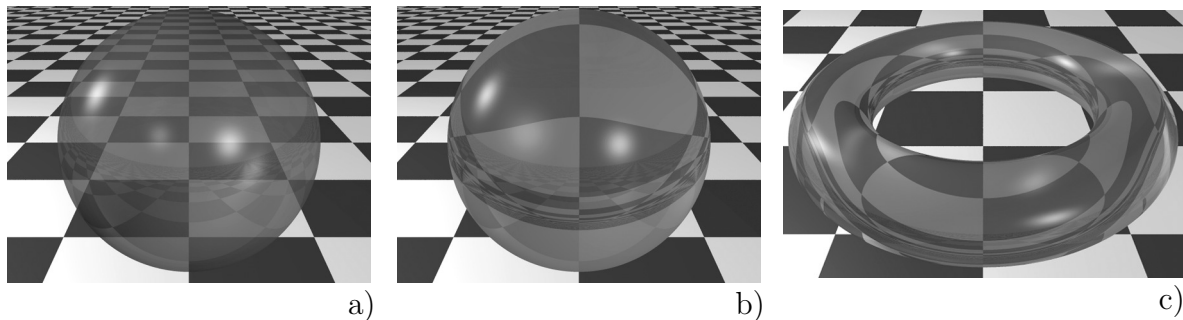


Abbildung 2.65: Transparenz ohne Brechung (a) und mit  $n = 1,5$  (b, c)



### 2.5.2 Lokale Beleuchtungsmodelle, Beleuchtungskomponenten, Simulation von Oberflächenstrukturen

Die Beleuchtung von Gegenständen in der realen Natur kann durch das Reflexionsgesetz allein nicht beschrieben werden – nur wenige Gegenstände verhalten sich wie Spiegel. Neben der direkten Reflexion haben weitere Beleuchtungskomponenten eine Bedeutung. Lokale Beleuchtungsmodelle beschreiben die Helligkeiten bzw. Farben<sup>108</sup> von Punkten auf Objektoberflächen in Abhängigkeit vom direkten Lichteinfall einer oder mehrerer Lichtquellen. Bei der Berechnung des Gesamtbildes sind dann die hier für einzelne Punkte geschilderten Berechnungen für alle Oberflächenpunkte, die auf Pixel des Bildes projiziert bzw. von verfolgten Lichtstrahlen getroffen werden, auszuführen.

#### Ambiente Beleuchtung (Umgebungslicht)

Die ambiente (richtungsunabhängige) Beleuchtung wird in der Natur durch vielfältige Lichtstreuungen an Teilchen in der Atmosphäre hervorgerufen und tritt insbesondere bei Nebel stark auf. Die Intensität  $I$  der ambienten Beleuchtung hängt nur von der globalen Intensität  $I_a$  des ambienten Lichtes und einem Faktor  $k_a$  ab, der angibt, wie stark die Oberfläche des betrachteten Körpers ambientes Licht wiedergibt:

$$I = I_a \cdot k_a \quad \text{mit} \quad 0 \leq k_a \leq 1. \quad (2.18)$$

Die ambiente Beleuchtung ist somit völlig unabhängig von der gegenseitigen Lage von Objekt, Kamera und Lichtquellen. Ein räumlicher Eindruck entsteht bei ausschließlich ambienter Beleuchtung nicht, eine Kugel erscheint wie in Abb. 2.61.

#### Diffuse Reflexion (Streulicht)

Oberflächen wie weißes Papier, die nicht sehr glatt sind, reflektieren Licht unabhängig vom Einfallswinkel in alle Richtungen. Dies kommt dadurch zustande, dass die Lichtstrahlen unterschiedlich tief in unregelmäßig geformte Oberflächen eindringen (siehe Abb. 2.66) und in verschiedene Richtungen zurückgeworfen werden. Je rauer eine Oberfläche ist, desto unterschiedlicher werden die Strahlen reflektiert bzw. gestreut.



Abbildung 2.66: Diffuse Reflexion

Um die so entstehende diffuse Reflexion zu beschreiben, wird die Leuchtdichte  $I_j$  des von einer Lichtquelle ausgehenden Lichtes betrachtet. Die Leuchtdichte beschreibt, wie viel Lichtleistung eine Lichtquelle pro Flächeneinheit abgibt. Tritt das Licht in einem flachen Winkel auf eine Oberfläche auf, so wird die Lichtleistung auf eine recht große Fläche verteilt. Die Intensität der Beleuchtung wird dadurch geringer. Wenn Licht, das durch ein Flächenstück  $A$  senkrecht zur Ausbreitungsrichtung scheint, auf ein Flächenstück  $A'$  der Materialoberfläche verteilt wird (siehe Abb. 2.67), so gilt für die Beleuchtungsichte  $I$  der Oberfläche:

<sup>108</sup>Die folgenden Betrachtungen werden lediglich anhand von Helligkeits- bzw. Intensitätswerten geführt. Durch die Durchführung der beschriebenen Berechnungen für die Intensitäten jeder der drei Grundfarben Rot, Grün und Blau lassen sich damit beliebige Farben darstellen.

$$I = \frac{A}{A'} \cdot I_j.$$

Das Verhältnis der Flächeninhalte  $A$  und  $A'$  lässt sich durch den Einfallswinkel  $\varphi$  ausdrücken:

$$A = A' \cdot \cos \varphi,$$

wobei für  $\varphi$  Werte von 0 bis  $90^\circ$  betrachtet werden. Für Licht, das auf die Rückseite auftrifft ( $\varphi > 90^\circ$ ), würden sich negative Werte ergeben, was natürlich nicht sinnvoll ist. Für die Beleuchtungsintensität  $I$ , die das Licht auf der Oberfläche hervorruft, gilt:

$$I = \frac{A}{A'} \cdot I_j = I_j \cdot \cos \varphi.$$

Der Einfallswinkel kann als Skalarprodukt des normierten Lichteinfallsvektors  $\vec{l}$  mit dem Normaleneinheitsvektor  $\vec{n}$  im betrachteten Punkt angegeben werden:

$$\cos \varphi = \langle \vec{n}, \vec{l} \rangle.$$

Damit lässt sich die Beleuchtungsintensität, die durch diffuse Reflexion des Lichtes einer Lichtquelle mit der Lichtintensität  $I_j$  in einem Punkt verursacht wird, durch das LAMBERTSche Kosinusgesetz ausdrücken:

$$\begin{aligned} I &= I_j \cdot k_d \cdot \max(0, \cos \varphi) \\ &= \begin{cases} I_j \cdot k_d \cdot \langle \vec{n}, \vec{l} \rangle & \text{falls } \langle \vec{n}, \vec{l} \rangle > 0 \\ 0 & \text{falls } \langle \vec{n}, \vec{l} \rangle \leq 0. \end{cases} \quad (2.19) \end{aligned}$$

Dabei ist  $k_d$  (mit  $0 \leq k_d \leq 1$ ) eine Materialkonstante, die angibt, wie stark eine Oberfläche Licht diffus reflektiert. Der diffuse Anteil des von einer Lichtquelle auf eine Oberfläche fallenden Lichtes ist somit abhängig von der Lichteinfallsrichtung, aber unabhängig von der Richtung zum Betrachter.

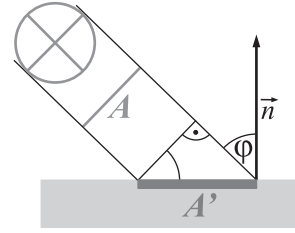


Abbildung 2.67: Auf ein Flächenstück auftreffendes Licht

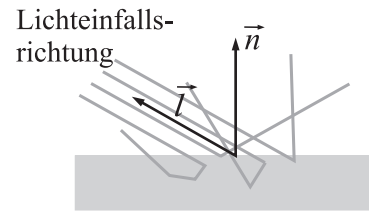


Abbildung 2.68: Lichteinfalls- und Normaleneinheitsvektor

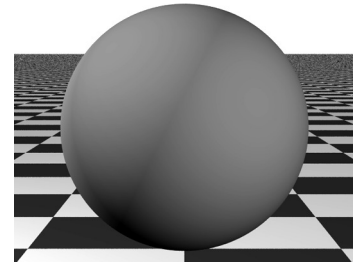
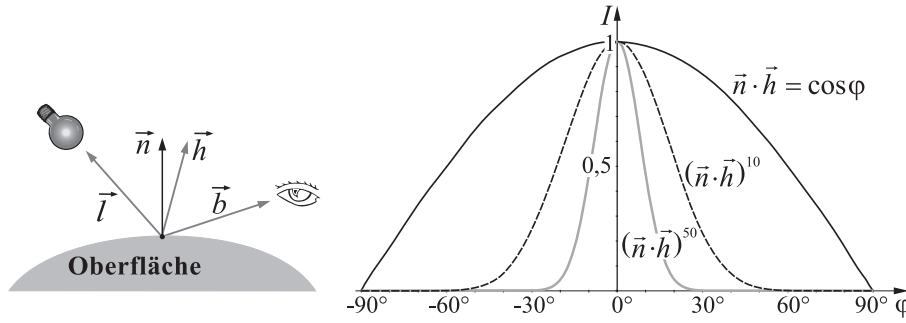


Abbildung 2.69: Diffuse Beleuchtung

### Entstehung von Glanzpunkten (spekulare Highlights)

Da reale Oberflächen keine idealen Spiegel sind, beschreibt das Reflexionsgesetz die Realität nur sehr unvollständig. Dennoch tritt auch bei nicht spiegelnden Oberflächen eine als *spekular* bezeichnete Beleuchtungskomponente auf, die vom Blickwinkel des Betrachters abhängt und somit nicht durch diffuse Reflexion erklärt werden kann. Diese Komponente ist durch Leuchtflecken („Highlights“) auf Körperoberflächen wahrzunehmen – gut sichtbar z. B. auf Billardkugeln. Da Lichtquellen i. Allg. als punktförmig angenommen werden, entstehen derartige Leuchtflecken durch direkte Reflexion nicht.

Das PHONGSche Modell der spekularen Reflexion generiert mehr oder weniger scharf abgegrenzte Leuchtbereiche in der Nähe der Punkte, in denen die Lichtquellen reflek-

Abbildung 2.70:  
Highlight-VektorAbbildung 2.71:  
Einfluss des Rauigkeitsindex auf den Verlauf von Glanzpunkten

tiert werden. Dazu wird ein *Highlight-Vektor*  $\vec{h}$  betrachtet (siehe Abb. 2.70), der die Richtung der Winkelhalbierenden der Verbindungsvektoren des Oberflächenpunktes zur Lichtquelle und zur Kamera angibt:

$$\vec{h} = \frac{\vec{b} + \vec{l}}{|\vec{b} + \vec{l}|}.$$

Wenn der Highlight-Vektor mit dem Normaleneinheitsvektor im betrachteten Flächenpunkt zusammenfällt, wird das Licht der Lichtquelle direkt zur Kamera reflektiert. Das Skalarprodukt der beiden Einheitsvektoren ist dann Eins. Je größer der Winkel zwischen Highlight-Vektor und Normaleneinheitsvektor ist, desto weniger beleuchtet die Lichtquelle den betrachteten Punkt der Oberfläche und das Skalarprodukt wird immer kleiner. Die Intensität der PHONGschen Beleuchtung durch eine Lichtquelle mit der Lichtintensität  $I_j$  wird durch die folgende Gleichung beschrieben:

$$I = I_j \cdot k_p \cdot \max \left( 0, \langle \vec{n}, \vec{h} \rangle^m \right) = \begin{cases} I_j \cdot k_p \cdot \langle \vec{n}, \vec{h} \rangle^m & \text{falls } \langle \vec{n}, \vec{h} \rangle > 0 \\ 0 & \text{falls } \langle \vec{n}, \vec{h} \rangle \leq 0. \end{cases} \quad (2.20)$$

Dabei ist  $k_p$  (mit  $0 \leq k_p \leq 1$ ) eine Materialkonstante, die angibt, wie stark Glanzpunkte auf einer Oberfläche hervortreten. Der Exponent  $m$  (in POV-Ray `phong_size`) wird als *Rauigkeitsindex* bezeichnet. Bei großen Rauigkeitsindizes fällt die Helligkeit bei wachsendem Winkel zwischen  $\vec{n}$  und  $\vec{h}$  schneller ab (siehe Abb. 2.71); es entstehen somit kleinere und schärfer abgegrenzte Glanzpunkte.

Oberflächen aus Metall weisen kleine und stark hervortretende Glanzpunkte auf; sie zeichnen sich somit durch hohe Werte des Rauigkeitsindex  $m$  aus. In Abb. 2.72 sind Beispiele für  $m = 10$  (a) und  $m = 50$  (b, c) dargestellt, wobei der metallische Anschein der Kugel in Abb. 2.72 c) durch die Kombination kleiner Glanzpunkte mit direkter Reflexion hervorgerufen wird (siehe auch S. 353 in Anhang B).

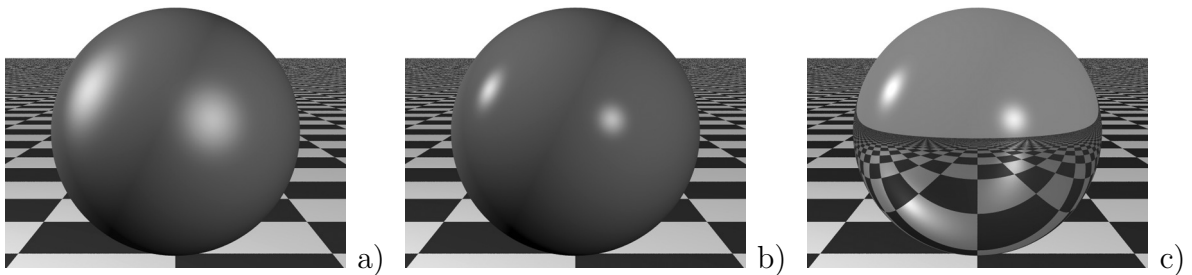


Abbildung 2.72: Größe der Glanzpunkte bei verschiedenen Rauigkeitsindizes

## Zusammenfassung der Beleuchtungskomponenten

Die durch die Gleichungen 2.18, 2.19 und 2.20 beschriebenen ambienten, diffusen und spekularen (PHONGschen) Beleuchtungskomponenten lassen sich zu folgender Gleichung für die Lichtintensität  $I$  eines Punktes einer Oberfläche zusammenfassen:

$$I = k_a I_a \cdot O + k_d \cdot \sum_{j=1}^n I_j \cdot \max\left(0, \langle \vec{n}, \vec{l}_j \rangle\right) \cdot O + k_p \cdot \sum_{j=1}^n I_j \cdot \max\left(0, \langle \vec{n}, \vec{h}_j \rangle\right)^m. \quad (2.21)$$

Dabei ist  $n$  die Anzahl der Lichtquellen in der Szene mit den Intensitäten  $I_1, \dots, I_n$ ;  $I_a$  die Intensität des ambienten Lichtes;  $\vec{n}$  der Normaleneinheitsvektor im betrachteten Flächenpunkt;  $\vec{l}_j$  der normierte Verbindungsvektor zur  $j$ -ten Lichtquelle und  $\vec{h}_j$  der Highlight-Vektor in Bezug auf diese Lichtquelle. Durch die Faktoren  $k_a$  (**ambient**),  $k_d$  (**diffuse**),  $k_p$  (**phong**) und den Rauigkeitsindex  $m$  (**phong\_size**)<sup>109</sup> sowie die in Abschnitt 2.5.1 definierte Intensität  $k_s$  der direkten Reflexion<sup>110</sup> wird die Beschaffenheit einer Oberfläche beschrieben. Der Faktor  $O$  steht für die Farbe der Oberfläche.<sup>111</sup>

Das durch Gleichung (2.21) beschriebene lokale Beleuchtungsmodell wurde 1975 von PHONG BUI-TUONG vorgeschlagen und wird daher oft als *PHONGsches Beleuchtungsmodell* bezeichnet. Es stellt eine Erweiterung des LAMBERTschen *Beleuchtungsmodells* dar, welches der Gleichung (2.21) unter Weglassung des letzten Summenterms entspricht, somit also nur die ambiente und diffuse Beleuchtung beschreibt und keine Leuchtpunkte generiert. Dieses Modell kommt aufgrund des geringeren Rechenaufwands dann zur Anwendung, wenn eine sehr schnelle Bildberechnung notwendig ist.

Durch das Verhältnis der beschriebenen Beleuchtungskomponenten und die Festlegung der Farbe lässt sich das Aussehen der Oberfläche eines Körpers in recht weiten Grenzen bestimmen. In POV-Ray wird dazu dem entsprechenden Objekt durch den folgenden Anweisungsblock eine Oberflächenerscheinung (Textur) zugewiesen:

```
texture {pigment {color rgbft < r, g, b, k_f, k_t > }
      finish {ambient k_a diffuse k_d phong k_p phong_size m reflection k_s} }.112
```

Das PHONGsche Beleuchtungsmodell ist durch vielfältige Erweiterungen und noch realitätsnähere Modellierungen ergänzt worden (siehe z. B. [11], S. 288–290 und [257],

<sup>109</sup>Durch die in Klammern angegebenen Begriffe können die Werte in POV-Ray festgelegt werden.

<sup>110</sup>Die direkte Reflexion bleibt in Gleichung (2.21) jedoch unberücksichtigt, da diese nicht nur durch Lichtquellen, sondern vor allem durch andere Objekte hervorgerufen wird und somit kein Bestandteil eines lokalen Beleuchtungsmodells ist. (Dies gilt ebenso für die Transparenz bzw. Brechung.) Einfache und schnelle Rendering-Verfahren berechnen keine Reflexionen anderer Objekte. Das in Abschnitt 2.5.5 beschriebene Raytracing-Verfahren berücksichtigt direkte Reflexionen hingegen prinzipbedingt.

<sup>111</sup>Es handelt sich bei diesem Faktor um die der betrachteten Grundfarbe entsprechende Komponente des Farbvektors, der aus den Komponenten Rot, Grün und Blau besteht. Wie bereits erwähnt wurde, muss die Lichtintensität für jede der drei Grundfarben des RGB-Modells (siehe 2.1.2) berechnet werden.

<sup>112</sup>Abb. B.18 auf S. 353 zeigt eine Kugel, deren Oberfläche mit dieser Definition und den dort jeweils angegebenen Werten für  $k_a$ ,  $k_d$ ,  $k_p$ ,  $m$ ,  $k_s$ , sowie Transparenzwerten  $k_f$  und  $k_t$  (siehe S. 130) erzeugt wurde. Mithilfe der Datei *Beleuchtungskomponenten.pov* in dem Ordner 2-5 (siehe S. 365), mit der die Abbildungen generiert wurden, lassen sich beliebige Kombinationen dieser Parameter testen.

S. 242–253). So verfeinerte BLINN 1977 das von PHONG entwickelte Modell der spekularen Highlights. Mit der spektralen Zusammensetzung von Glanzlichtern in Abhängigkeit vom Lichteinfallswinkel und vom Material beschäftigten sich COOK and TORRENCE 1982 – im Modell von PHONG haben Glanzlichter grundsätzlich die Farbe der Lichtquelle. Es handelt sich dabei um recht subtile Verfeinerungen, deren Wirkung oft kaum sichtbar ist, weshalb das beschriebene PHONGsche Modell noch immer häufig angewendet wird. Andere Untersuchungen betreffen die Mikrogeometrie von Oberflächen und ermöglichen die überzeugende Simulation von Rauigkeit durch unterschiedliche Ausrichtungen von Mikrofacetten. In POV-Ray steht diese Möglichkeit durch Einfügen der Anweisung `crand` (mit einem Wert von 0 bis 1) zur Verfügung. Auch andere Erweiterungen des hier beschriebenen Beleuchtungsmodells können durch die Parameter `specular`, `roughness`, `shininess`, `metallic` u. a. genutzt werden (siehe [200], S. 179–185) und treten mit teilweise anderen Bezeichnungen auch in den Texturierungsmodulen anderer Grafikprogramme (siehe 2.7.3) auf. Als erweiterte Beleuchtungsmodelle seien schließlich noch anisotropische Shader (siehe z. B. [49], S. 95–99) genannt, welche die Abhängigkeit der Lichtreflexion von der Richtung auf einer Oberfläche berücksichtigen und damit z. B. gebürstetes Metall simulieren können.

Der Simulation von Oberflächenstrukturen dienen in der fotorealistischen Computergrafik nicht nur lokale Beleuchtungsmodelle. So können Vertiefungen und Erhebungen, die durch Schatten und veränderte Glanzlichter wahrnehmbar sind, durch die Beeinflussung der Lichtreflexion mittels *Bump Mapping*, d. h. die Projektion spezieller Bilddateien („Einbuchtungskarten“ bzw. Bump Maps) vorgetäuscht werden (siehe z. B. [189], S. 266–273). Die Projektion von Bump Maps auf Oberflächen entspricht der in Abschnitt 2.4.6 beschriebenen Texturprojektion. Abb. 2.73 b) zeigt eine Kugel, auf die außer einer Karte der Erde (wie bereits in Abb. 2.59 auf S. 126) noch die in Abb. 2.73 a) dargestellte Bump Map projiziert wurde; eine farbige Version enthält S. 354.

Das Verfahren des Bump Mapping hat sich in breitem Umfang durchgesetzt. Noch realistischer (und entsprechend rechenaufwändiger) sind Verfahren, bei denen Maps die Geometrie der Objekte verändern (*Displacement Mapping*, siehe z. B. [49], S. 83–95).

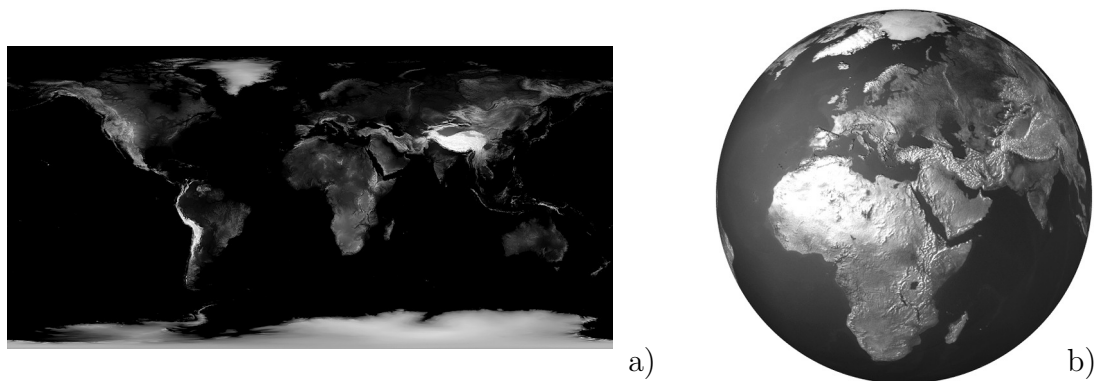


Abbildung 2.73: Erde mit Höhenprofil (Bump Map)

### 2.5.3 Interpolation von Farben oder Oberflächennormalen

Wie in den vorangegangenen Abschnitten deutlich wurde, haben für die Erscheinung von Oberflächen deren Normalen eine entscheidende Bedeutung, da die meisten Beleuchtungskomponenten anhand der Normaleneinheitsvektoren bestimmt werden.

Die am häufigsten auftretende Repräsentation geometrischer Objekte in der 3D-Computergrafik ist die durch Dreiecksnetze (siehe Abschnitt 2.3.9). Da sich beim Übergang von einem Dreieck zu einem anderen die Richtung der Normalen abrupt ändert, entstehen harte Farbübergänge, wie bereits in Abb. 2.42 auf S. 109 anhand einer triangulierten konischen Spirale deutlich wurde. Als weiteres Beispiel zeigt Abb. 2.74 das in POV-Ray berechnete Bild eines aus etwa 6300 Dreiecken bestehenden Fisches, dessen Facetten deutlich erkennbar sind. Um diesen meist unerwünschten Effekt zu beseitigen, ist entweder eine Verfeinerung der Dreiecksnetze<sup>113</sup> oder eine Interpolation von Helligkeiten bzw. Facettennormalen innerhalb der Dreiecke notwendig. Das Ziel einer solchen Interpolation besteht darin, dass benachbarte Dreiecke in gemeinsamen Kanten oder Ecken gleiche Helligkeitswerte annehmen. Dazu wird jedem Eckpunkt eines Dreiecks ein Einheitsvektor  $\vec{n}_m$  zugeordnet, der durch Berechnung des normierten Mittelwertes der Normaleneinheitsvektoren  $\vec{n}_i$  aller  $k$  an diesem Punkt anliegenden Dreiecke entsteht:

$$\vec{n}_m = \frac{\sum_{i=1}^k \vec{n}_i}{\left| \sum_{i=1}^k \vec{n}_i \right|}$$

(siehe Abb. 2.75). Jedes Dreieck wird somit durch die drei Eckpunkte und die ihnen zugeordneten Normaleneinheitsvektoren beschrieben, in der Syntax von POV-Ray:

```
smooth_triangle {  <x1,y1,z1>,<xn1,yn1,zn1>,
                   <x2,y2,z2>,<xn2,yn2,zn2>,
                   <x3,y3,z3>,<xn3,yn3,zn3>  }.
```

Bei der Bildberechnung bestehen zwei Möglichkeiten, die Eckennormalen für eine Glättung der Helligkeitsübergänge zwischen Kanten einzusetzen. Bei dem 1971 erstmals

<sup>113</sup>Eine solche Verfeinerung lässt sich durch Unterteilungsalgorithmen herbeiführen, die auf S. 111 beschrieben wurden (Subdivision Surfaces). Allerdings erzeugt eine Unterteilung, die keine Kanten mehr sichtbar werden lässt, eine extrem große Zahl von Dreiecken, was zu einem sehr hohen Speicherbedarf und zu langen Renderzeiten führt. Aus diesem Grunde werden zur Kantenglättung vorwiegend die in diesem Abschnitt beschriebenen Interpolationsverfahren verwendet.

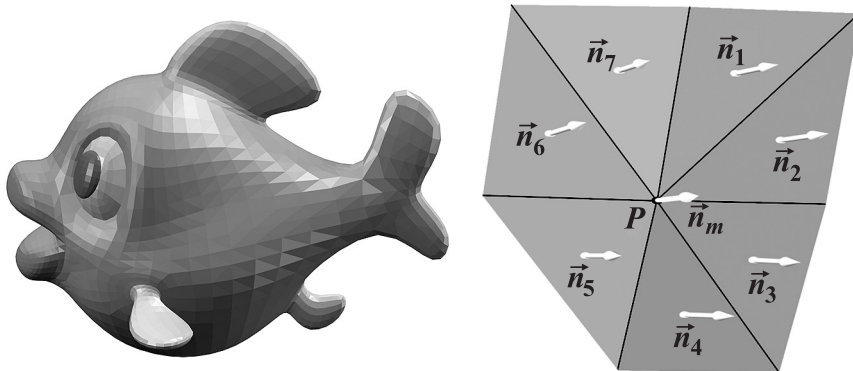


Abbildung 2.74: Facetten eines Fisches (Flat Shading)

Abbildung 2.75: Eckpunktnormale des Fisches

vorgeschlagenen GOURAUD-Shading wird jedem Eckpunkt anhand des zugehörigen Normaleneinheitsvektors für jede der drei Grundfarben ein Helligkeitswert zugeordnet.

Innerhalb der Facetten erfolgt eine Interpolation der Helligkeitswerte. Die Position eines Punktes  $P$  innerhalb eines Dreiecks  $\triangle ABC$  kann dazu durch die Koeffizienten  $s$  und  $t$  mit

$$P = A + s \cdot \overrightarrow{AB} + t \cdot s \cdot \overrightarrow{BC} \quad (0 \leq s, t \leq 1)$$

ausgedrückt werden (siehe Abb. 2.76). Sind  $I_A, I_B, I_C$  die Helligkeitswerte der Punkte  $A, B$  und  $C$ , so erhält  $P$  die Helligkeit

$$I_P = (1-s) \cdot I_A + (1-t) \cdot s \cdot I_B + t \cdot s \cdot I_C.$$

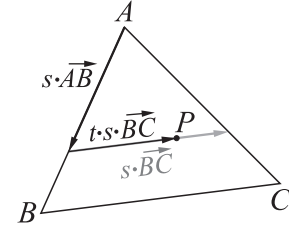


Abbildung 2.76: Interpolationskoeffizienten

In der Praxis wird die Helligkeitsinterpolation oft erst nach der Projektion in die Bildebene inkrementell entlang von Rasterzeilen ausgeführt (vgl. [125], S. 310ff. und [189], S. 189ff.), wodurch der Berechnungsaufwand geringer ist. GOURAUD-Shading kommt daher oft für schnelle Anwendungen (z. B. die Echtzeit-Vorschau in 3D-Grafiksystemen) zum Einsatz. Für fotorealistisches Rendering ist die Qualität dieses Verfahrens nur dann ausreichend, wenn die Objekte sehr kleine (und entsprechend viele) Facetten haben. Durch die lineare Interpolation und die stetigen Helligkeitsübergänge an den Kanten entstehen zwar stetige Helligkeitsverläufe auf Oberflächen; die ersten Ableitungen sind jedoch i. Allg. nicht stetig. Deren Unstetigkeitsstellen an den Kanten nimmt das Auge oft wahr, da es Kanten mit schwachen Kontrastübergängen verstärkt.<sup>114</sup>

Ein meist noch schwerer wiegendes Defizit der GOURAUD-Schattierung besteht darin, dass Glanzpunkte nicht hinreichend genau berechnet werden können (siehe Abb. 2.77). Liegt nämlich ein Glanzpunkt innerhalb einer Facette und ist er gegenüber der Facette recht klein, so wird dieser Glanzpunkt nicht erzeugt, da die Helligkeiten der Eckpunkte geringer sind als die Helligkeit, die der Glanzpunkt haben sollte.

Erfolgt anstelle der beschriebenen Helligkeitsinterpolation eine *lineare Interpolation der Normaleneinheitsvektoren*, indem jedem betrachteten Punkt durch

$$\vec{n}_P = (1-s) \cdot \vec{n}_A + (1-t) \cdot s \cdot \vec{n}_B + t \cdot s \cdot \vec{n}_C$$

eine „eigene“ Normale (in Abhängigkeit von seiner durch die Koeffizienten  $s$  und  $t$  ausgedrückten Lage innerhalb einer Facette) zugeordnet wird, so erfolgt eine korrekte Darstellung von „Highlights“ (siehe Abb. 2.78). Eine auf diese Weise berechnete Normale eines inneren Punktes einer Facette kann mit dem Highlight-Vektor (siehe S. 133) durchaus einen kleineren Winkel bilden als jede der drei Eckpunktnormalen und dadurch nach Gleichung (2.20), S. 133, heller ausgeleuchtet werden.

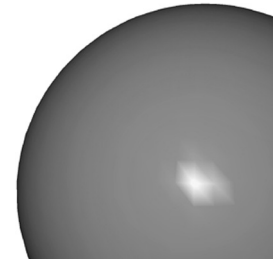


Abbildung 2.77: GOURAUD-Shading

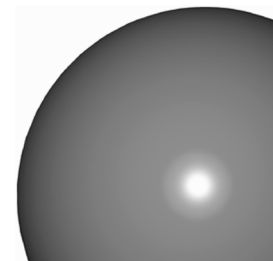


Abbildung 2.78: PHONG-Shading

<sup>114</sup>Diese Eigenschaft des menschlichen Auges wurde 1865 von ERNST MACH entdeckt und wird als MACH-Band-Effekt bezeichnet (siehe z. B. [31], S. 138f. und [79], S. 558).



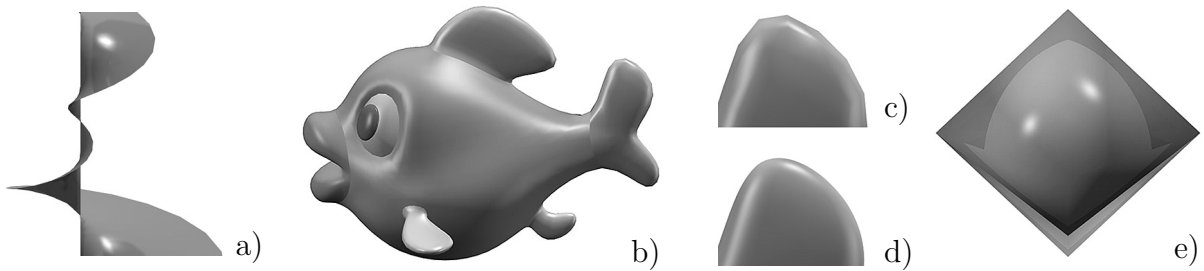


Abbildung 2.79: PHONG-Shading (a, b, c, e) sowie Subdivision Surface (d)

Ein auf diese Weise arbeitendes Verfahren wurde 1975 von PHONG BUI-TUONG vorgeschlagen und ist als PHONG-Shading bekannt (siehe u. a. [79], S. 560ff., [125], S. 313f. und [257], S. 208f.). Es benötigt erheblich mehr Rechenleistung als die Helligkeitsinterpolation nach GOURAUD.<sup>115</sup> Es kommt daher bei Anwendungen zum Einsatz, die weniger eine Minimierung der Renderzeit als eine hohe Qualität der berechneten Bilder erfordern, so z. B. in teilweise etwas modifizierter Form in Raytracing-Software.

Die auf S. 136 beschriebene Berechnung der Eckpunktnormalen wird durch die meisten 3D-Grafikprogramme intern vorgenommen. In POV-Ray müssen die Normaleneinheitsvektoren jedoch in der Form `smooth_triangle` (siehe S. 136) oder in der `mesh2`-Beschreibung (siehe S. 109) im Zusammenhang mit den zugehörigen Eckpunkten angegeben werden, wobei Hilfsprogramme existieren, welche die Eckpunktnormalen von Dreiecksnetzen berechnen.<sup>116</sup> Das auf S. 109 erwähnte Paket SurfaceGraphics3D für Mathematica kann Eckpunktnormalen generieren und „geglättete Dreiecksnetze“ für die Verwendung in POV-Ray exportieren; auf diese Weise entstand Abb. 2.79 a).<sup>117</sup> Der in Abb. 2.79 b) dargestellte Fisch hat dieselben Eckpunkte wie derjenige in Abb. 2.74 auf S. 136. Bei genauer Betrachtung (insbesondere der vergrößerten Schwanzflosse in Abb. 2.79 c) fällt auf, dass auch der glatte Fisch an seinen Außenkanten etwas eckig wirkt<sup>118</sup> – seine Geometrie ist trotz der durch die PHONG-Schattierung bewirkten Kantenglättung mit der des ursprünglichen Fisches identisch. Zum Vergleich wurde die untere Schwanzflosse (Abb. 2.79 d) durch dreifache Flächenunterteilung als Subdivision Surface (siehe S. 111) erzeugt – mit einer um ein Vielfaches höheren Berechnungszeit.

Einen Extremfall der Kantenglättung durch Normaleninterpolation zeigt Abb. 2.79 e). Hierbei wurde ein Oktaeder durch die `smooth_triangle`-Beschreibung dargestellt, was natürlich zu Darstellungsfehlern führt.<sup>119</sup>

<sup>115</sup>Die Zeit sparende Interpolation der Helligkeitswerte nach der Projektion auf das Bildfenster, also im Zweidimensionalen, kommt für die Interpolation der Normalen natürlich nicht in Frage, diese muss im dreidimensionalen Raum erfolgen. Die für die Beleuchtung der Bildpunkte nach Gl. (2.21) auf S. 134 benötigten Skalarprodukte sind dabei für jeden Punkt, der auf ein Pixel des Bildfensters abgebildet wird, unter Verwendung des ihm durch die Interpolation zugeordneten Normalenvektors zu berechnen.

<sup>116</sup>Derartige Hilfsprogramme sind z. B. PoseRay (<http://mysite.verizon.net/sfg0000/>) und das ältere 3ds2pov, mit dem die Eckennormalen des in Abb. 2.79 abgebildeten Fisches berechnet wurden.

<sup>117</sup>Vgl. dazu die bis auf die Eckpunktnormalen auf identische Weise erzeugte Abb. 2.42 auf S. 109.

<sup>118</sup>Größere und farbige Versionen der Abbildungen befinden sich auf S. 354 in Anhang B.

<sup>119</sup>Siehe dazu auch das Video `Oktaeder-Glatt-Video.mpg` (siehe S. 365).



### 2.5.4 Scanline Rendering

In den vorangegangenen Abschnitten wurden alle notwendigen Schritte beschrieben, die für die Erzeugung eines Abbildes einer räumlichen Szene notwendig sind: die Projektion in eine Bildebene unter Einschränkung auf ein Sichtfenster, die Entfernung nicht sichtbarer Objekte bzw. Facetten sowie die Zuweisung von Helligkeits- bzw. Farbwerten zu Punkten auf Objektoberflächen. Zur letztendlichen Bilderzeugung werden für alle Pixel des Sichtfensters (siehe Abb. 2.56 auf S. 123) die Farbwerte berechnet. Diese Berechnung wird i. Allg. zeilenweise durchgeführt und deshalb als „Scanline Rendering“ (Bildberechnung durch Zeilenabtastung) bezeichnet. Abhängig von der Art der Projektion, den verwendeten Sichtbarkeitsuntersuchungen (siehe 2.4.4 und 2.4.5), Schattierungsverfahren (Flat-, GOURAUD- oder PHONG-Shading) und den berücksichtigten lokalen Beleuchtungsmodellen (LAMBERTSches oder PHONGSches Modell, siehe S. 134) können Scanline-Verfahren sehr unterschiedlich aufgebaut sein und sich in der Qualität der erzeugten Bilder sowie in der Berechnungsgeschwindigkeit erheblich unterscheiden.

Soll für eine sehr schnelle Bildberechnung die Schattierung (nach dem GOURAUD-Verfahren) erst nach der Projektion in der Ebene des Bildes erfolgen, so sind alle geometrischen Objekte in Polygonnetze umzuwandeln. Erfolgt die Berechnung der Helligkeitswerte von Oberflächenpunkten im Raum, so können Körper und Flächen auch exakt abgebildet werden. In diesem Falle ist die Berechnung von Schnittpunkten und Normalen anhand der analytischen Darstellungen der Objekte erforderlich, was meist einen hohen Rechenaufwand verursacht. In den weitaus meisten Fällen erfolgt deshalb bei Scanline-Rendering-Verfahren eine Triangulation aller geometrischen Objekte.

Seit der Mitte der neunziger Jahre wurde die Mehrzahl der für das Scanline-Rendering notwendigen Berechnungen in spezielle Grafikprozessoren, die Bestandteile von Grafikkarten sind, ausgelagert.<sup>120</sup> Dazu gehören die Sichtbarkeitsbestimmung nach dem *z*-Buffer-Verfahren und die Schattierung sowie seit dem Ende der neunziger Jahre auch die vor der Projektion notwendige Ansichtstransformation und das Clipping. Mit der Implementation in Grafikhardware konnte die Darstellungsgeschwindigkeit von 3D-Szenen um mehrere Größenordnungen gesteigert werden. Wurden derartige „3D-Beschleuniger“ anfangs vor allem auf (sehr teuren) Workstations für die Echtzeit-Darstellung komplexer Konstruktionen in CAD-Software genutzt, gehören sie mittlerweile zur normalen Ausstattung von Personalcomputern und kommen u. a. für die Darstellung der verbreiteten 3D-Computerspiele zum Einsatz. Moderne Grafikprozessoren haben inzwischen die Komplexität von Hauptprozessoren (CPUs) erreicht. Ein Ende der Entwicklung hin zu immer leistungsfähigeren Bausteinen ist Anfang 2006 nicht in Sicht.<sup>121</sup>

<sup>120</sup>Siehe dazu u. a. [18], S. 202–207, [19], S. 194–201 und [51], S. 130–135.

<sup>121</sup>Zum einen steigt mit jeder neuen Chipgeneration die Anzahl der Polygone, die Grafikkarten in jeder Sekunde verarbeiten können. Eine andere Entwicklungslinie besteht in der Perfektionierung von Shadern und Texturierungsalgorithmen, die mittlerweile auch für in Echtzeit ablaufende Computerspiele eine hohe Darstellungsqualität erreichen. Dazu werden auch ständig leistungsfähigere Anti-Aliasing-Algorithmen (siehe Abschnitt 2.5.7) sowie Verfahren für die Filterung von Texture Maps (siehe 2.4.6) implementiert.

### 2.5.5 Verfolgung von Lichtstrahlen (Raytracing)

Den im vorangegangenen Abschnitt beschriebenen Scanline-Rendering-Verfahren liegen ausschließlich lokale Beleuchtungsmodelle zugrunde. Daher berücksichtigen diese keine Reflexionen und Lichtbrechungen und können auch Schatten nicht physikalisch korrekt berechnen.<sup>122</sup> Für die Erzeugung fotorealistischer dreidimensionaler Darstellungen hoher Qualität hat daher das Raytracing-Verfahren eine besondere Bedeutung erlangt, welches die Lichtausbreitung simuliert. Es wird – zunehmend ergänzt um Verfahren der globalen Beleuchtung (siehe Abschnitt 2.5.6) – von der überwiegenden Mehrzahl der aktuellen 3D-Softwarepakete angewendet.<sup>123</sup>

In der Realität nimmt das Auge Lichtstrahlen wahr, die von Lichtquellen ausgehen und das Auge auf direktem, vor allem aber auf indirektem Wege erreichen. Auf ihrem Wege werden die Lichtstrahlen meist an verschiedenen Gegenständen reflektiert. Eine Grafiksoftware kann nicht die Verläufe aller von den Lichtquellen ausgehenden Lichtstrahlen berechnen – das würde extrem lange dauern und die meisten dieser Lichtstrahlen erreichen das Auge nie (auch nicht indirekt), haben also keinen Einfluss auf das Bild. Nach einem Gesetz der Optik sind jedoch Verläufe von Lichtstrahlen umkehrbar. Diese Tatsache wird bei dem Raytracing-Verfahren (übersetzt: Strahlverfolgung) genutzt. Vom „Auge des Betrachters“ bzw. von der virtuellen Kamera aus werden die Wege von Lichtstrahlen – möglicherweise über mehrfache Reflexionen und Lichtbrechungen hinweg – verfolgt (siehe Abb. 2.80 auf S. 141). Raytracing-Programme berechnen dadurch automatisch Schatten und Spiegelungen.

Das Raytracing-Verfahren beinhaltet eine Zentralprojektion, da – ausgehend vom Projektionszentrum – nach Objekten „gesucht“ wird, die auf den betrachteten Ausschnitt der Projektionsebene abgebildet werden. Für jeden Punkt des zu berechnenden Bildes muss eine Raytracing-Software Helligkeits- bzw. Farbwerte berechnen und dazu den Verlauf eines Sehstrahls verfolgen. Die Anzahl der zu verfolgenden Lichtstrahlen ist von der gewünschten Auflösung des Bildes abhängig, bei  $800 \times 600$  Pixeln müssen z. B. 480 000 Strahlen verfolgt werden. Jeder dieser Strahlen kann durch Spiegelungen und Transparenz mehrfach verzweigen, wie bereits das einfache Beispiel in Abb. 2.80 zeigt.

Für jeden zu verfolgenden Lichtstrahl sind folgende Schritte durchzuführen:

1. Bestimmung des vom Sichtfenster aus nächstgelegenen Schnittpunktes des Strahls mit einer Objektoberfläche. Ein derartiger Schnittpunkt wird *Knoten des Sichtstrahlenbaums* genannt. In dem Knoten muss der orientierte Normaleneinheitsvektor der Fläche berechnet werden.
2. Für den Lichtstrahl werden in dem unter 1. ermittelten Schnittpunkt unter Berücksichtigung der geometrischen und optischen Eigenschaften des Objekts die

---

<sup>122</sup>Allerdings erfolgen zum Teil sehr echt wirkende Imitationen durch die auf S. 130 erwähnten Maps.

<sup>123</sup>Die in der vorliegenden Arbeit häufig betrachtete Software POV-Ray basiert auf dem Raytracing-Verfahren; auch die bereits erwähnten Programme 3ds max und Carrara nutzen es für hochwertiges Rendering, wobei für die schnellere Bildberechnung zusätzlich Scanline-Renderer zur Verfügung stehen.

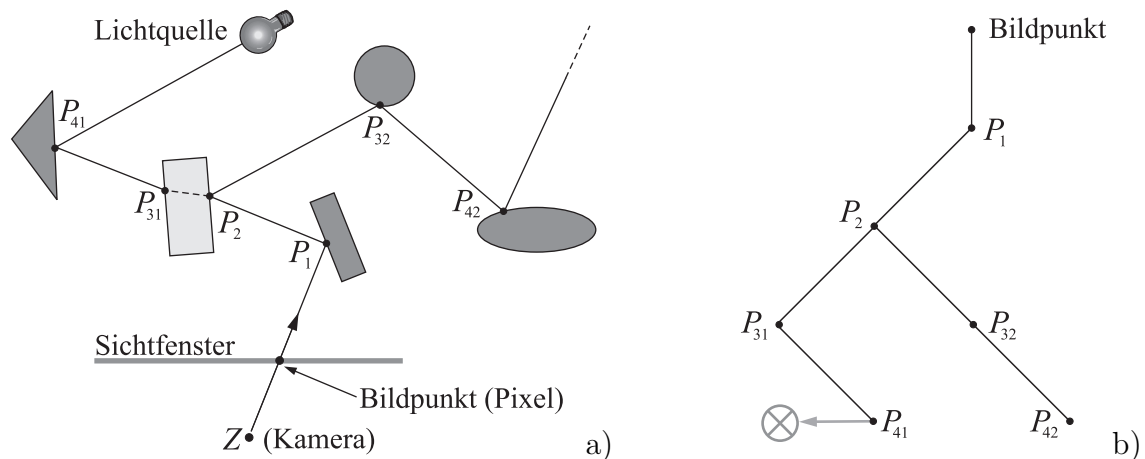


Abbildung 2.80: a) Strahlverfolgung bei einer einfachen Szene; b) Sichtstrahlenbaum

Richtungen sowie farbabhängigen Intensitäten reflektierter und bei transparenten Objekten transmittierter und dabei evtl. gebrochener Lichtstrahlen berechnet.

3. Für jeden reflektierten oder transmittierten Lichtstrahl werden erneut die Schritte 1. und 2. durchgeführt, wodurch neue Knoten und Lichtstrahlen entstehen, für die wiederum 1. und 2. auszuführen sind. Da dieser Prozess unendlich lange dauern könnte, wird das Verfahren nach einer festgelegten Zahl von Reflexionen oder einer festgelegten Abschwächung, die der Strahl durch Reflexionen bzw. Durchdringungen von Körpern erfahren hat, abgebrochen.<sup>124</sup>
4. Für jeden Knoten des Sichtstrahlenbaums wird ein Helligkeitswert errechnet, der sich aus dahinter liegenden Reflexionen bzw. Durchdringungen und der unmittelbaren Beleuchtung durch die Lichtquellen nach einem lokalen Beleuchtungsmodell (siehe Abschnitt 2.5.2) ergibt. Im Sichtstrahlenbaum weiter hinten liegende Knoten haben Einfluss auf das Erscheinungsbild weiter vorn liegender Knoten, da diese das Licht der hinten liegenden Knoten zur Kamera hin reflektieren bzw. brechen. Jedem Knoten wird (für jede Grundfarbe) ein Helligkeitswert zugeordnet, der sich aus seiner lokalen Beleuchtung und (gewichtet mit Faktoren, welche die Abschwächung der Helligkeit durch Reflexion bzw. Durchdringung beschreiben) den Helligkeitswerten der in Reflexions- bzw. Lichtbrechungsrichtung unmittelbar hinter ihm liegenden Knoten ergibt.
5. Die Helligkeits- bzw. Farbwerte der von der Kamera aus ersten Knoten werden den Pixeln des Bildes zugeordnet.

Raytracing ist, wie aus diesen Schritten deutlich wird, ein rekursives Verfahren. Die farbabhängige Helligkeit in einem Punkt einer Oberfläche (bzw. einem Knoten des Sichtstrahlenbaums) kann nach der Gleichung

<sup>124</sup>Zum möglichen Einfluss der Strahlverfolgungstiefe auf das berechnete Bild siehe Abb. B.23 auf S. 355. Abb. 2.80 zeigt die von einem einzigen Bildpunkt ausgehenden Strahlen und Knoten bei einer Strahlverfolgungstiefe von 4 für eine einfache Szene. Zur Vereinfachung wurde dabei angenommen, dass an den Innenflächen des teiltransparenten Körpers keine Reflexionen auftreten.

$$I = k_a I_a \cdot O + \sum_{j=1}^n I_j \cdot \frac{s_j}{d_j^2} \left( k_d \cdot \max \left( 0, \langle \vec{n}, \vec{l}_j \rangle \right) \cdot O + k_p \cdot \max \left( 0, \langle \vec{n}, \vec{h}_j \rangle \right)^m \right) + k_s I(P_r) + k_t I(P_t) \quad (2.22)$$

berechnet werden, die einige Gemeinsamkeiten mit der Gleichung (2.21) auf S. 134 für die lokale Beleuchtung durch  $n$  Lichtquellen nach dem PHONGschen Beleuchtungsmodell aufweist. Die Bedeutung der Lichtintensitäten  $I_a, I_1, \dots, I_n$ , der Vektoren  $\vec{n}, \vec{l}_j, \vec{h}_j$  sowie der Koeffizienten  $O, k_a, k_d, k_p$  und  $m$  wurde bereits auf S. 134 dargelegt. Zusätzlich beschreibt der Faktor  $s_j / d_j^2$  die Verdeckung (für  $s_j = 0$ ) von Lichtquellen durch undurchsichtige Objekte bzw. deren Abschwächung (für  $0 < s_j < 1$ ) durch transparente Objekte sowie die in Abhängigkeit von der Entfernung  $d$  des beleuchteten Punktes zur Lichtquelle erfolgende Abschwächung der Lichtintensität.<sup>125</sup> Die Erzeugung von Schatten ist somit ein Bestandteil des Raytracing-Verfahrens, kann jedoch durch Setzen von  $s_j = 1$  auch unterlassen werden. Viele Programme bieten die Möglichkeit, Schattenwurf für einzelne Objekte zu deaktivieren.<sup>126</sup>

Der bedeutendste Unterschied zwischen den Gleichungen (2.21) für die lokale Beleuchtung und (2.22) für die Beleuchtung nach dem Raytracing-Verfahren besteht in den Summanden  $k_s I(P_r)$  und  $k_t I(P_t)$ . Die Intensitäten  $I(P_r)$  und  $I(P_t)$  sind nämlich die wiederum nach Gleichung (2.22) zu ermittelnden Farbintensitäten in den (in Reflexions- bzw. Refraktionsrichtung) nächsten Knoten  $P_r$  bzw.  $P_t$  des Sichtstrahlenbaums<sup>127</sup> (siehe Abb. 2.80). Diese bestimmen (gewichtet durch die Faktoren  $k_s$  und  $k_t$  für die Intensitäten der direkten Reflexion bzw. die Transparenz) die Helligkeit des jeweils betrachteten Punktes mit. Gleichung (2.22) ermöglicht somit nicht die explizite Berechnung der Farbintensitäten in einem Knoten, sondern gibt lediglich an, wie diese rekursiv bestimmt werden kann. Der dafür notwendige Rechenaufwand wird anhand der Verzweigungsstruktur des Sichtstrahlenbaums deutlich und wächst bei komplexen Szenen mit vielen transparenten Objekten exponentiell mit der Strahlverfolgungstiefe.<sup>128</sup>

Wurden die auf S. 140–141 beschriebenen Schritte 1.–5. für jedes Pixel des Sichtfensters mit der vorgegebenen Strahlverfolgungstiefe ausgeführt, so ist die Bildberechnung abgeschlossen. Das Raytracing-Verfahren beinhaltet auch die notwendigen Sichtbarkeitsuntersuchungen, da in Schritt 1 für jeden Lichtstrahl der Schnittpunkt mit dem vom

<sup>125</sup>Die Energie des Lichtes einer punktförmigen Lichtquelle, das auf eine Flächeneinheit trifft, nimmt mit dem Quadrat des Abstandes zwischen Lichtquelle und Fläche ab.

<sup>126</sup>In POV-Ray geschieht dies durch Einfügen der Anweisung `no_shadow` in die Objektbeschreibung. Insbesondere für Illustrationen ist es aus Gründen der Übersichtlichkeit oft sinnvoll, Schatten zu deaktivieren. Für viele der Abbildungen in dieser Arbeit wurde daher der Schattenwurf unterdrückt.

<sup>127</sup>Die Helligkeiten bzw. Farbintensitäten dieser Knoten hängen ihrerseits von den Intensitäten der von ihnen aus nächsten Knoten ab. Diese werden durch die Intensitäten der eine weitere Reflexion oder Brechung entfernten Knoten beeinflusst und so weiter.

<sup>128</sup>In der Praxis werden Lichtstrahlen meist über 5–10 Reflexionen bzw. Durchdringungen verfolgt. Danach ist die Lichtintensität i. Allg. derart abgeschwächt, dass ein Abbrechen der Strahlverfolgung kaum noch einen Einfluss auf das Ergebnis hat. In POV-Ray lässt sich die Rekursionstiefe der Strahlverfolgung mittels `max_trace_level` einstellen. Beispiele für Bilder derselben Szene, die mit unterschiedlichen Strahlverfolgungstiefen gerendert wurden, befinden sich auf S. 355 in Anhang B.

Sichtpunkt bzw. Anfangsknoten aus nächsten Schnittpunkt mit einem Objekt ermittelt werden muss. Bei Szenen mit vielen Objekten kann dies allerdings sehr zeitaufwändig sein. Aus diesem Grunde werden in der Praxis mittels grober Begrenzungskörper (meist Quader oder Kugeln, siehe dazu auch S. 95) Voruntersuchungen ausgeführt, welche Objekte Strahlen überhaupt schneiden können, oder es erfolgt eine Unterteilung des Raumes mithilfe von Octree-Verfahren (siehe z. B. [102], S. 107ff.).

Das Raytracing-Verfahren ermöglicht die direkte Verarbeitung geometrischer Objekte ohne die Umwandlung in Polygonnetze und damit einhergehende Ungenauigkeiten.<sup>129</sup> Die Schnittpunkte von Lichtstrahlen mit Ebenen, dreieckigen Facetten und Quadern können durch Lösen linearer Gleichungssysteme bestimmt werden. Für Kugeln, Kegel, Zylinder und Flächen zweiter Ordnung ist das Lösen quadratischer Gleichungssysteme notwendig.<sup>130</sup> Schnittpunktberechnungen von Strahlen mit komplexeren Objekten wie Isosurfaces und Parameterflächen erfordern wesentlich aufwändigere numerische Verfahren (siehe S. 95–97). Um für derartige Objekte Normaleneinheitsvektoren zu bestimmen, müssen auch die partiellen Ableitungen meist mittels numerischer Näherungsverfahren ermittelt werden,<sup>131</sup> was ebenfalls einen hohen Berechnungsaufwand mit sich bringt. Die Wahl günstiger Repräsentationen geometrischer Objekte kann somit aufgrund des unterschiedlichen Aufwands für die Berechnung von Schnittpunkten und Normalen wesentlich zur Einsparung von Rechenzeit beitragen.

Das in Abb. 2.81 dargestellte Hyperboloid wurde als **quadric**-Objekt (in POV-Ray implementierter Objekttyp, der Flächen 2. Ordnung mithilfe quadratischer Gleichungssysteme behandelt) in 2 Sekunden gerendert. Exakt dasselbe Bild entstand durch die Berechnung des Hyperboloids als **isosurface**-Objekt (gegeben durch die Gleichung  $x^2 + y^2 - z^2 - 0.5 = 0$ ) in 8 Sekunden, da POV-Ray durch Gleichungen gegebene Objekte mittels numerischer Näherungsverfahren auswertet.

Die Messungen wurden mit den Dateien

- `Hyperboloid-quadric.pov` und
- `Hyperboloid-isosurface.pov`,

(Ordner 2–5, siehe S. 365) durchgeführt.

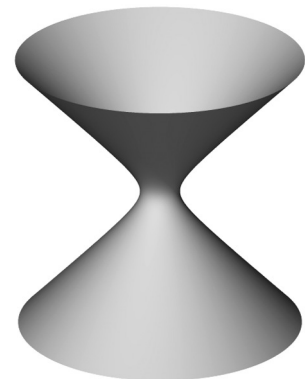


Abbildung 2.81: Hyperboloid

<sup>129</sup>POV-Ray verhält sich in dieser Hinsicht besonders „konsequent“ und verarbeitet alle geometrischen Objekte bis hin zu Parameterflächen anhand ihrer analytischen Beschreibungen (mit Ausnahme von Objekten, die der Software bereits als Dreiecksnetze übergeben werden). Der damit verbundene Rechenaufwand schränkt allerdings die Verwendbarkeit von POV-Ray für die Videoproduktion ein, bei der für jede Filmsekunde 25–30 Bilder zu generieren sind. Daher greifen andere Raytracing-Programme stärker auf die Repräsentation durch Polygonnetze zurück, welche schneller verarbeitet werden können.

<sup>130</sup>Auch die notwendige Bestimmung der Normaleneinheitsvektoren der Flächen in den Schnittpunkten (Knoten) ist direkt anhand der geometrischen Eigenschaften bzw. der analytischen Beschreibungen dieser Klassen von Objekten möglich.

<sup>131</sup>Die beiden Normaleneinheitsvektoren in einem Punkt einer Fläche sind senkrecht zur Tangentialebene an die Fläche in diesem Punkt und errechnen sich deshalb für eine Fläche mit der Parametendarstellung  $\vec{x}(u, v)$  durch  $\vec{n}_{1,2} = \pm \frac{\frac{\partial \vec{x}}{\partial u} \times \frac{\partial \vec{x}}{\partial v}}{\left| \frac{\partial \vec{x}}{\partial u} \times \frac{\partial \vec{x}}{\partial v} \right|}$ .

Grundsätzlich ermöglicht das Raytracing-Verfahren aufgrund der physikalisch korrekten Simulation von Schatten, Spiegelungen und Lichtbrechungen sowie der Behandlung geometrischer Objekte anhand ihrer analytischen Beschreibungen Ergebnisse sehr hoher Qualität, die oft tatsächlich als „fotorealistisch“ zu bezeichnen sind. Es erkauft dies mit einem hohen Rechenaufwand, der für die Verwendung unter Produktionsbedingungen Optimierungen<sup>132</sup> und Kompromisse<sup>133</sup> erfordert. Für die Berechnung einfacher Szenen mit Vorschauqualität genügen allerdings auf im Jahr 2006 aktuellen PCs meist wenige Sekunden, auch höher aufgelöste Bilder sind oft nach weniger als einer Minute gerendert. Die Erzeugung von Videos mit 1500 Bildern für jede Minute Videomaterial ist jedoch immer noch ein recht langwieriger Vorgang. In Filmstudios kommen dazu „Renderfarmen“ zum Einsatz, bei denen Bildberechnungen auf teilweise mehrere Hundert Hochleistungscomputer verteilt werden.

Trotz seiner Möglichkeit, hochwertige fotorealistische Darstellungen zu erzeugen, weist das Raytracing-Verfahren in der hier beschriebenen Form einige Defizite auf, die durch Modifikationen und Ergänzungen beseitigt oder zumindest gemildert werden können. Dazu gehören vor allem folgende Probleme:

- Da alle Farbwerte der Pixel berechneter Bilder aus den separat ermittelten Intensitäten der drei Grundfarben bestimmt werden, sind zwar Farbverschiebungen bei Reflexionen oder Lichtbrechungen in dem Sinne möglich, dass Körper bestimmte der Grundfarben stärker oder schwächer reflektieren bzw. hindurch lassen. Eine tatsächliche Wellenlängenabhängigkeit der Reflexion oder der Lichtbrechung besteht jedoch nicht.<sup>134</sup> Bei den in computergrafischen Darstellungen mitunter zu findenden Regenbögen handelt es sich meist um separat erzeugte, teilweise transparente geometrische Objekte, die in die entsprechenden Szenen eingefügt wurden.
- Aufgrund der Punktförmigkeit der Lichtquellen haben Schatten grundsätzlich harte Begrenzungen. Dem ist dadurch zu begegnen, dass ausgedehnte Lichtquellen durch mehrere rasterförmig angeordnete Punktlichter simuliert werden (siehe

---

<sup>132</sup>Zu Optimierungen von Raytracing-Algorithmen siehe u. a. [94] und [102].

<sup>133</sup>Ein möglicher Kompromiss ist die Kombination von Scanline Rendering und Raytracing. Der für Großprojekte oft genutzte Renderer Mental Ray (siehe [www.mentalimages.com](http://www.mentalimages.com)) berechnet aus Geschwindigkeitsgründen zunächst rechteckige Bildausschnitte nach einem Scanline-Algorithmus. Immer dann, wenn eine Richtungsänderung von Lichtstrahlen aufgrund von Reflexionen oder Brechungen erfolgt, wechselt dieses Programm zum Raytracing-Verfahren (siehe [49], S. 14ff.). Außerdem kann es für die Schattenerzeugung Shadow-Maps (ebd., S. 138ff.) verwenden, die zwar optisch nicht exakt, dafür aber schneller zu berechnen sind. Die Rendermodule der 3D-Grafikprogramme 3ds max und Maxon Cinema 4D ermöglichen ebenfalls die Wahl zwischen Shadow-Maps und Raytraced Shadows.

<sup>134</sup>Eine Möglichkeit, das Raytracing-Verfahren durch ein wellenlängenabhängiges Reflexionsmodell zu ergänzen, wird in [248], S. 142 beschrieben. Auch (physikalisch nicht exakte) Simulationen von Dispersionserscheinungen sind in einigen Raytracing-Programmen möglich (siehe z. B. [200], S. 237). Dazu werden diskretisierte Spektren von Lichtstrahlen mit unterschiedlichen Farben erzeugt, denen verschiedene Ausbreitungsgeschwindigkeiten (und somit unterschiedliche Brechungsindizes, siehe S. 129) in verschiedenen Medien zugeordnet sind. Um auf diese Weise die Erzeugung eines Farbspektrums an einem Prisma zu simulieren, sind sehr hohe Anzahlen von Strahlverläufen zu berechnen; daraus ergeben sich Renderzeiten von mehreren Stunden bis hin zu Tagen auf 2005/06 gebräuchlichen PCs.

z. B. [49], S. 117–142, [200], S. 150–153 und [282], S. 327). Da für die Auswertung von Gleichung (2.22) auf S. 142 bei einer höheren Anzahl  $n$  von Lichtquellen mehr Zeit benötigt wird, können durch ausgedehnte Lichtquellen bewirkte weiche Schatten die Bildberechnung erheblich verlangsamen (siehe Abb. B.24 auf S. 355 und die dazu angegebenen Renderzeiten).

- Die Umkehrung der Strahlenrichtung ermöglicht zwar die exakte Modellierung von Reflexionen und Lichtbrechungen. Lichtstrahlen werden jedoch nur berücksichtigt, wenn sie letztendlich den Betrachter erreichen. Durch die dadurch bedingte „passive Rolle“ der Lichtquellen erfolgt keine Darstellung von Lichtbündelungen (Kautiken), die z. B. durch Linsen und Hohlspiegel hervorgerufen und auf in Brennpunktnähe befindlichen Objekten sichtbar werden.
- Farben von Objekten erkennt der Beobachter nach dem Prinzip des Raytracing nur dann in anderen Objekten wieder, wenn eine direkte Reflexion erfolgt. In der Realität beeinflussen Gegenstände aber auch durch diffuse Reflexion von Licht das Aussehen nahe gelegener anderer Gegenstände. Die diffuse Beleuchtung in lokalen Beleuchtungsmodellen und bei dem Raytracing-Verfahren ist rein passiv und addiert lediglich Helligkeitswerte zu Punkten auf Objektoberflächen. Die diffuse Beleuchtung von Objekten durch andere Objekte wird dadurch nicht beschrieben.

Die beiden zuletzt genannten Probleme können durch die Kombination von Raytracing mit Verfahren der globalen Illumination gelöst werden.

### 2.5.6 Globale Beleuchtungsmodelle (Radiosity)

In der Realität entsteht ein großer Teil der Beleuchtung durch indirekte (diffuse) Mehrfachreflexionen, wodurch Objekte durch das von anderen Objekten reflektierte Licht sichtbar werden. So erfolgt eine Beleuchtung von Gegenständen in einem Raum durch das von matten Wänden reflektierte Licht.<sup>135</sup> Umgekehrt hat z. B. eine weiße Wand in der Nähe eines großen roten Gegenstandes einen rötlichen „Schimmer“, denn Körper strahlen einen Teil des Lichtes, das sie von Lichtquellen empfangen, in diffuser Weise wieder ab – wobei das Licht die Körperfarbe annimmt – und beleuchten dabei indirekt ihre Umgebung (siehe Abb. B.25 auf S. 356).

Unter den Modellen, die Mehrfachreflexionen von Licht zwischen verschiedenen Flächen und die dadurch hervorgerufene Beleuchtung der Flächen beschreiben, hat das *Radiosity-Verfahren*<sup>136</sup> eine besondere Bedeutung erlangt. Sein Ansatz ist die Erhaltung

---

<sup>135</sup>Ein Raum ohne Lampe wird durch das zum Fenster einfallende Licht auch dann beleuchtet, wenn sich keine direkt reflektierenden Gegenstände, sondern nur matte Wände in dem Raum befinden. Durch lokale Beleuchtungsmodelle und Raytracing kann dieser Tatsache nur durch die ambiente Beleuchtungskomponente (siehe S. 131) entsprochen werden, die aber alle Bereiche des Raumes völlig undifferenziert ausleuchtet, was der Realität ebenso widerspricht wie eine Ausleuchtung nur der Bereiche, die von den durch das Fenster einfallenden Sonnenstrahlen direkt getroffen werden.

<sup>136</sup>Eine Grundlage für die Entwicklung von Radiosity-Verfahren bildeten Modelle der Wärmetechnik, die entwickelt wurden, um die Strahlungsübertragung von Wärme zwischen Oberflächen zu simulieren.

der Lichtenergie in einer geschlossenen Szene. Bei diesem Verfahren muss zunächst die gesamte Szene in eine endliche Zahl  $n$  diskreter Flächenstücke (Patches) zerlegt werden, von denen angenommen wird, dass sie über ihre gesamten Flächen gleichmäßig Licht abstrahlen und reflektieren. Trifft auf ein Flächenstück mit dem Index  $i$  Licht der Energie  $E$ , so wird ein Teil  $E_{Ri} = r_i \cdot E$  dieser Energie reflektiert;  $r_i$  mit  $0 \leq r_i \leq 1$  ist dabei der Reflexionsfaktor des Flächenstücks. Der Radiosity-Ansatz geht davon aus, dass jedes Objekt auch selbst Licht ausstrahlen (emittieren) kann. Somit ist die Strahlungsenergie (kurz als Strahlung oder Radiosity bezeichnet) eines Flächenstücks die Summe der von dem Flächenstück selbst emittierten und der von ihm reflektierten Anteile der Strahlung, die es von allen vorhandenen Flächenstücken empfängt.<sup>137</sup>

$$E_i = E_{Ei} + r_i \sum_{j=1}^n E_j F_{ij}. \quad (2.23)$$

Dabei ist  $E_{Ei}$  die von dem betrachteten Flächenstück (mit dem Index  $i$ ) emittierte Eigenstrahlung,  $E_j$  die von dem Flächenstück mit dem Index  $j$  abgegebene Strahlungsenergie sowie  $F_{ij}$  der Anteil an der von Fläche  $j$  abgegebenen Energie, die auf Fläche  $i$  auftrifft, genannt *Formfaktor*. Dieser Faktor hängt von den geometrischen Formen der beteiligten Flächenstücke sowie ihrer gegenseitigen Lage und Sichtbarkeit ab.

Um die Strahlungsenergien aller Flächenstücke zu berechnen, ist das durch (2.23) gegebene lineare Gleichungssystem, bestehend aus  $n$  Gleichungen in den Variablen  $E_1, \dots, E_n$ , zu lösen. Dazu müssen die Emissionsenergien  $E_{E1}, \dots, E_{En}$ , die Reflexionsfaktoren  $r_1, \dots, r_n$  sowie die Formfaktoren  $F_{ij}$  ( $i, j = 1 \dots n$ ) bekannt sein. Während es sich bei den Emissionsenergien und den Reflexionsfaktoren um Objekt- bzw. Oberflächeneigenschaften handelt, hängen die Formfaktoren von der Geometrie der Objekte und ihrer gegenseitigen Lage ab. Ihre Berechnung ist meist sehr aufwändig<sup>138</sup> und benötigt bei der Generierung von Grafiken mithilfe von Radiosity-Verfahren den größten Teil der Rechenzeit. Allerdings verursacht auch die Lösung des Gleichungssystems (2.23) einen hohen Aufwand. Um für komplexere Szenen eine realistische Ausleuchtung zu berechnen, ist eine Unterteilung in sehr viele Flächenstücke notwendig, so dass oft Gleichungssysteme mit mehr als 10 000 Gleichungen und Variablen entstehen.<sup>139</sup> Da

<sup>137</sup>Es ist auch möglich, den Strahlungsaustausch zwischen Flächen ohne die vorherige Zerlegung in eine endliche Zahl von Flächenstücken zu beschreiben. Anstelle von Gleichung (2.23) ergibt sich dann eine Integralgleichung. Deren Lösung ist jedoch nur in wenigen Spezialfällen möglich, weshalb in der Praxis immer eine Diskretisierung durch Zerlegung in eine endliche Zahl von Flächenstücken vorgenommen wird. Diese Diskretisierung und die damit verbundene Herleitung von (2.23) aus einer allgemeineren Radiosity-Gleichung in Integralform wird in [39], Kap. 3 beschrieben.

<sup>138</sup>Da die Beschreibung der verschiedenen Methoden der Formfaktorbestimmung erheblich den Rahmen der vorliegenden Arbeit überschreiten würde, sei hierzu auf [39], Kap. 4 verwiesen.

<sup>139</sup>Es kommen hierfür fast ausschließlich iterative Näherungsverfahren zum Einsatz, siehe z. B. [39], Kap. 5 und [257], S. 351–356. Vor der Lösung des Gleichungssystems hat die sinnvolle Aufteilung der Flächen in Patches eine hohe Bedeutung, damit möglichst wenige Flächenstücke (und somit relativ wenige Gleichungen und Variable) bei dennoch realistischer Beleuchtung entstehen. Dazu werden Verfahren der schrittweisen Verfeinerung bzw. adaptiven Flächenzerlegung in Abhängigkeit von der Änderungsrate der abgegebenen Strahlung verwendet (siehe u. a. [39], Kap. 6–8).



die Radiosity farbabhängig berechnet werden soll, um die Färbung von Objekten durch nahe gelegene Objekte zu beschreiben, sind diese Gleichungssysteme für jede der drei Grundfarben zu lösen.<sup>140</sup> Der Berechnungsaufwand für die globale Beleuchtung nach dem Radiosity-Verfahren ist somit hoch. Allerdings hängen sowohl die Formfaktoren als auch die Lösung des Gleichungssystems (2.23) nur von der Form und der gegenseitigen Lage der Lichtquellen und geometrischen Objekte ab. Sie müssen daher bei der Betrachtung von Szenen aus anderen Perspektiven nicht neu berechnet werden.<sup>141</sup>

Als universeller und einziger Ansatz für die Berechnung von Bildern ist das Radiosity-Verfahren in den meisten Fällen unbrauchbar, da es keine direkten Reflexionen und auch keine Lichtbrechung berücksichtigt. Aus diesem Grunde wird es in praktischen Anwendungen meist mit Raytracing kombiniert. Dabei erfolgt zunächst eine Ermittlung der globalen Beleuchtung durch Radiosity, wobei die zu den betrachteten Patches berechneten Strahlungsintensitäten gespeichert werden. Bei der anschließenden Berechnung des Bildes mithilfe des Raytracing-Verfahrens wird in jedem Knoten des Sichtstrahlenbaums die durch die Strahlungsintensität des Patches, auf dem der Knoten liegt, verursachte Helligkeit berücksichtigt.<sup>142</sup>

Ein Defizit des Radiosity-Verfahrens – auch in Kombination mit Raytracing – besteht darin, dass sich die Beleuchtung von Körpern durch andere Körper auf diffuse Reflexionen beschränkt. In der Realität treten auch andere Effekte indirekter Beleuchtung auf, wozu insbesondere die bereits auf S. 145 erwähnten *Kaustiken* zählen.

Kaustiken (meist auch in der deutschsprachigen Literatur als *Caustics* bezeichnet) entstehen aufgrund der Lichtbündelung durch gekrümmte Spiegel oder transparente Körper und machen sich als Leuchtflecken auf Oberflächen (wie in Abb. 2.82 auf dem Tisch, auf dem der Torus liegt) bemerkbar. Theoretisch wäre die Darstellung von Kaustiken mithilfe des Radiosity-Verfahrens möglich, wenn Flächen in extrem kleine Flächenstücken unterteilt würden. Dies ist jedoch aufgrund des dadurch verursachten Rechenaufwands bei der Bestimmung der Formfaktoren und der Lösung des Gleichungssystems (2.23) praktisch nicht realisierbar.

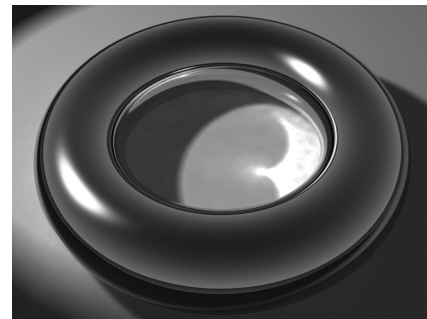


Abbildung 2.82: Caustics

Eine farbige Version dieser Abbildung befindet sich auf S. 356.

<sup>140</sup>Auf S. 356 in Anhang B bzw. in den entsprechenden Dateien im Ordner 2-5 (siehe Anhang D, S. 365) ist eine Szene, die mit und ohne Radiosity gerendert wurde, abgebildet. Die Färbung farbneutraler Wände durch zwei im Raum befindliche farbige Kugeln ist dabei deutlich erkennbar.

<sup>141</sup>Insbesondere für durch „Kamerafahrten“ erzeugte Animationen, die u. a. in der Architekturvisualisierung sehr gebräuchlich sind, genügt oft eine einmalige Berechnung der Formfaktoren und Strahlungsenergien, wodurch auch die Produktion längerer Videos mit globaler Beleuchtung möglich ist.

<sup>142</sup>Die Berücksichtigung von Radiosity führt durch die Addition von Helligkeitswerten generell zu helleren Szenen. Für die Erzeugung des Bildes ohne Radiosity auf S. 356 in Anhang B wurde deshalb eine zusätzliche Lichtquelle verwendet und außerdem die Intensität der in beiden Szenen vorhandenen Lichtquellen erhöht, um zu einem in der Helligkeit vergleichbaren Ergebnis zu gelangen.

Eine physikalisch korrekte Simulation sowohl von Kaustiken als auch der diffusen Reflexion ist durch die Kombination des in Abschnitt 2.5.5 beschriebenen Raytracing mit *Photon-Tracing* bzw. „Vorwärts-Raytracing“ möglich. Dabei werden von Lichtquellen emittierte „Photonen“<sup>143</sup> verfolgt. Diese können von Objekten reflektiert, absorbiert oder hindurchgelassen und dabei eventuell gebrochen werden, wobei sich ihre Richtung und Energie ändert.<sup>144</sup> Den von Photonen „getroffenen“ Punkten geometrischer Objekte werden Strahlungsenergien zugeordnet und in als *Photon-Maps* bezeichneten Datenstrukturen gespeichert. Nach dem Photon-Tracing erfolgt die Bildberechnung durch Raytracing. In jedem Knoten des Sichtstrahlenbaums wird dabei mittels der Photon-Maps die Dichte und Energie der Photonen in der Nähe des entsprechenden Punktes ermittelt. Durch Addition der so bestimmten indirekten Beleuchtungsintensitäten und der durch das Raytracing-Verfahren berechneten Lichtintensitäten in jedem Knoten ergibt sich die globale Beleuchtung einer Szene.

Anders als die Verläufe der von der Kamera ausgehenden Lichtstrahlen sind die Verläufe der Photonen und ihre an Punkte von Objektoberflächen abgegebenen Strahlungsenergien nicht von der Lage der Kamera abhängig. Es genügt daher für Animationen, bei denen sich nur die Richtung und Entfernung ändert, aus der eine Szene betrachtet wird, einmalig Photon-Maps zu berechnen. Es besteht hierbei also eine Gemeinsamkeit zwischen Photon-Tracing und dem Radiosity-Verfahren, welches bei Änderung der Perspektive ebenfalls keine Neuberechnung der Strahlungsenergien erfordert.

Das Photon-Tracing-Verfahren wurde Mitte der neunziger Jahre von HENRIK WANN JENSEN<sup>145</sup> entwickelt und ist somit jünger als das bereits Mitte der achtziger Jahre theoretisch ausgearbeitete Radiosity-Modell.<sup>146</sup> Zu den ersten Programmen, in welche die Photon-Tracing-Methode zur umfassenden Simulation globaler Illumination implementiert wurde, zählte Mental Ray.<sup>147</sup> Inzwischen verwenden auch andere Softwarepakete die Kombination von klassischem Raytracing und Photon-Tracing. Sowohl das Photon-Tracing- als auch das Radiosity-Verfahren werden beständig weiterentwickelt. Einen anwendungsorientierten Überblick über Verfahren und Effekte der globalen Beleuchtung (Radiosity, Caustics, Photon-Tracing) vermittelt [28], S. 238–247.

---

<sup>143</sup>Der Begriff „Photon“ ist aus der Physik bekannt und beschreibt dort ein Lichtquant (kleinstes Energieteilchen einer elektromagnetischen Strahlung). Auch die in der Computergrafik betrachteten Photonen simulieren die Übertragung von Strahlungsenergie. Jedoch ist Photon-Tracing keine quantenphysikalisch korrekte Simulation der Lichtausbreitung.

<sup>144</sup>Im Gegensatz zum klassischen (von der Kamera ausgehenden) Raytracing ist die Zahl der zu verfolgenden Photonen nicht durch die Auflösung des Bildes bestimmt, sondern muss anhand eines Kompromisses zwischen der gewünschten Genauigkeit der Beleuchtungssimulation und der benötigten Renderzeit festgelegt werden. Üblich ist die Verfolgung von zwischen einigen Tausend und mehreren Millionen Photonen. Um Abb. B.26 auf S. 356 zu generieren, wurden ca. 100 000 Photonen verwendet.

<sup>145</sup>Siehe die Internetseite <http://graphics.ucsd.edu/~henrik/>, auf der auch einige Publikationen zu den Themengebieten Photon-Tracing und Photon-Mapping zur Verfügung stehen.

<sup>146</sup>Allerdings konnte JENSEN auf bereits in den achtziger Jahren entwickelte Ansätze des stochastischen Raytracing zurückgreifen, siehe u. a. [94], Kap. 5.

<sup>147</sup>Siehe Abschnitt 2.7.2 sowie [49] (insbesondere S. 155–204) und [www.mentalimages.com](http://www.mentalimages.com).

### 2.5.7 Anti-Aliasing

Unabhängig von dem angewendeten Rendering-Verfahren erfolgt die Berechnung des Bildes einer dreidimensionalen Szene immer für eine diskrete Anzahl von Bildpunkten (Pixeln). Es wird somit eine *Rasterung* vorgenommen, die der auf S. 72 beschriebenen Rasterung zweidimensionaler Objekte für die Darstellung auf Bildschirmen entspricht. Bei der Berechnung von Bildern dreidimensionaler Szenen werden Pixeln Farben von Punkten auf Objektoberflächen zugeordnet, auf welche die verfolgten Strahlen treffen (bei Raytracing), bzw. welche auf die Zentren der entsprechenden Pixel projiziert werden (bei Scanline-Rendering).<sup>148</sup> An Kanten bzw. Kurven, welche die Bilder einzelner Objekte begrenzen, entstehen dabei Treppeneffekte (Aliasing), die bereits auf S. 73 beschrieben wurden. Abb. 2.84 a) auf S. 150 bzw. Abb. B.27 a) auf S. 356 zeigen diesen Effekt anhand des Bildes einer dreidimensionalen Szene.

Besonders problematisch stellen sich Aliasing-Erscheinungen dar, wenn Muster mit sich periodisch wiederholenden Strukturen auftreten, deren Abstände in der Projektion kleiner sind als das Doppelte des Abstandes zweier benachbarter Pixel. Durch die dabei auftretende Überlagerung der Muster mit dem Pixelraster entstehen Interferenzerscheinungen, die auch als *Moiré-Effekt* bezeichnet werden (siehe Abb. 2.83 a).<sup>149</sup>

Die vom Prinzip her einfachste Möglichkeit, Aliasing-Effekte zu vermindern, besteht in der Überabtastung (*Supersampling*); in der Computergrafik entspricht dies der Be-

<sup>148</sup>Durch die Rasterung erfolgt eine Diskretisierung kontinuierlicher Funktionen  $h(x, y, z)$  bzw.  $h(x, y)$  durch „Abtastung“, d. h. durch Bestimmung der Funktions- (Helligkeits- bzw. Farb-)werte für eine endliche Zahl von Punkten des Raumes bzw. einer Projektionsebene. Dieser Vorgang entspricht der Digitalisierung von Signalen, weshalb signaltheoretische Grundlagen für die Rasterung relevant sind (siehe z. B. [11], S. 311–330 und [122], S. 243–255). Häufig kommen zur Verminderung von Aliasing-Effekten Tiefpassfilter zum Einsatz, was die auf S. 69 erwähnten Fouriertransformation aus dem Orts- in den Frequenzbereich erfordert. Derartige Filterungen sind insbesondere für die Glättung mittels Texture-Mapping (siehe Abschnitt 2.4.6) auf dreidimensionale Objekte projizierter Pixelbilder erforderlich. Die Darstellung von signaltheoretischen Grundlagen und Verfahrensweisen bei der Bild- und Texturfilterung würde über den Rahmen der vorliegenden Arbeit erheblich hinausgehen. Für die Glättung von Kanten dreidimensionaler Objekte führen in den meisten Fällen die hier kurz dargestellten (vom Prinzip her elementarerem, aber sehr rechenaufwändigen) Vorgehensweisen zu guten Resultaten.

<sup>149</sup>Nach dem Nyquist-Shannon-Abtasttheorem treten Moiré-Muster bei einer Abtastung periodischer Strukturen mit kleineren Frequenzen als der doppelten Frequenz der Strukturen auf ([11], S. 316ff.).

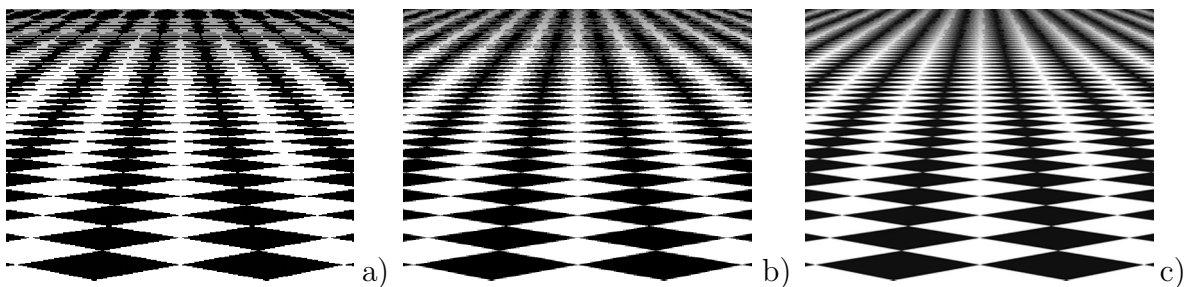


Abbildung 2.83: Aliasingeffekte an einem Schachbrettmuster

a) einfache Abtastung; b) zweifache Abtastung; c) zweifache Abtastung und Anti-Aliasing

rechnung einer höheren Zahl von Pixeln und einer anschließenden Skalierung (Bildverkleinerung) auf die gewünschte Anzahl von Pixeln.<sup>150</sup> So wurde Abb. 2.83 b) mit  $1182 \times 848$  Pixeln berechnet und anschließend auf  $591 \times 424$  Pixel (die Auflösung von Abb. 2.83 a) skaliert, wobei die Helligkeit jedes Pixels der Durchschnitt von vier berechneten Pixelhelligkeiten in der ursprünglichen Auflösung ist (vierfaches Supersampling). Der Nachteil dieses Verfahrens besteht in dem wesentlich erhöhten Rechenaufwand. Dabei bringt die Berechnung einer höheren Zahl von Pixeln innerhalb farblich homogener Flächen keinen Nutzen. Viele 3D-Grafikprogramme bieten daher die Möglichkeit, *adaptive Anti-Aliasing* durchzuführen. Dazu wird das Bild zunächst mit der gewünschten Auflösung berechnet, wobei die Software die Helligkeits- bzw. Farbwerte benachbarter Pixel vergleicht. Treten dabei Differenzen oberhalb eines festgelegten Schwellenwertes auf, so wird zwischen zwei benachbarten Pixeln ein weiterer Punkt abgetastet. Die Qualität von Abb. 2.83 b) lässt sich auf diese Weise mit geringerem Aufwand erreichen als mit globalem vierfachem Supersampling.

Bei *rekursivem adaptivem Anti-Aliasing* werden die Helligkeiten zusätzlich berechneter Punkte erneut mit denen der benachbarten Abtastpunkte verglichen und bei Überschreiten des Schwellenwertes nochmals Helligkeitswerte für dazwischen liegende Punkte berechnet, bis eine festgelegte Rekursionstiefe erreicht ist. Da die menschliche Wahrnehmung Störungen, die als regelmäßige Muster auftreten, stärker wahrnimmt als zufällig verteilte Bildfehler, empfiehlt es sich, die Abtastpunkte bei Supersampling oder adaptivem Anti-Aliasing nicht in exakt gleichen Abständen zu benachbarten Pixeln anzuordnen, sondern mithilfe einer Zufallskomponente geringfügig zu streuen. Ein solches Vorgehen wird als *Jittering* bezeichnet (siehe z. B. [248], S. 146f.) und trägt vor allem recht wirksam zur Reduzierung von Moiré-Mustern bei.

Die Abbildungen 2.83 c), 2.84 b) sowie B.27 b) auf S. 356 entstanden in POV-Ray unter Nutzung von rekursivem adaptivem Anti-Aliasing (Schwellenwert 10% der maximalen Helligkeit, Rekursionstiefe 3) mit einer Zufallskomponente bei der Verteilung der Abtastpunkte (Jitter mit maximaler Versetzung der Abtastpunkte von den Mittelpunkten um 50% des Abstandes zu benachbarten Abtastpunkten):

```
Antialias_Depth=3
Antialias_Threshold=0.1
Jitter_Amount=0.5 .
```

<sup>150</sup>Die „Herunterskalierung“ entspricht signaltheoretisch einer Tiefpassfilterung.

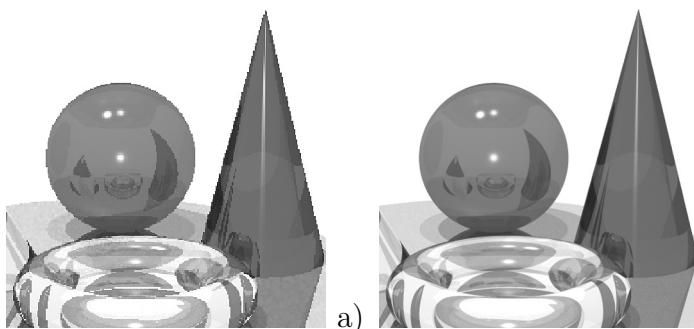


Abbildung 2.84: Ausschnitt aus einer Szene: a) ohne Anti-Aliasing; b) mit Anti-Aliasing

Die Berechnung der gesamten Szene (siehe Abb. B.27 auf S. 356) benötigte mit einer Auflösung von jeweils  $800 \times 600$  Pixeln 9 Sekunden ohne und 23 Sekunden mit Anti-Aliasing auf einem PC mit Pentium IV (3,2 GHz).

## 2.6 Animationen

Bei Animationen (Filme, die durch Trickfilmtechniken oder mithilfe des Computers erzeugt werden) handelt es sich um Sequenzen (Folgen) vieler einzelner Bilder (meist als *Frames* bezeichnet), die sich von Bild zu Bild nur geringfügig ändern, so dass für den Betrachter scheinbar kontinuierliche Bewegungen oder sonstige Änderungen sichtbar sind. Um die Illusion von Kontinuität hervorzurufen, sind mindestens ca. 25 Bilder je Sekunde (FPS – Frames Per Second) erforderlich. Zwischen den einzelnen Bildern können sich prinzipiell alle Inhalte und Bildeigenschaften verändern:

- Positionen, Sichtbarkeit und Formen von Objekten,
- Eigenschaften von Lichtquellen und Objektoberflächen (Helligkeiten und Farben),
- Position, Blickpunkt sowie Öffnungswinkel der Kamera und somit die Sicht auf die Szene.

### 2.6.1 Animationen auf Grundlage von Parameterdarstellungen

Um Computeranimationen zu erzeugen, ist es notwendig, Bestimmungsgrößen (Koordinaten, Transformationen, Farben u. a.) zeitabhängig zu beschreiben.<sup>151</sup> Prinzipiell lässt sich jede Eigenschaft, die durch Zahlenwerte darstellbar ist, animieren, d. h. zeitabhängig verändern. Die entsprechenden Werte sind dazu als Funktionen eines Parameters, der die Zeit beschreibt, auszudrücken. Die meisten 3D-Grafikprogramme enthalten Animationsmodule, die zeitabhängige Beschreibungen durch Interpolation von Positionen und Werten, die der Anwender für bestimmte „Schlüsselbilder“ vorgibt, generieren. In skriptgesteuerten Grafikprogrammen sowie in CAS sind hingegen Parameterdarstellungen der zu animierenden Größen in Abhängigkeit von einem Parameter (in POV-Ray `clock`) explizit anzugeben.<sup>152</sup> Im Folgenden werden anhand einiger Beispiele Möglichkeiten beschrieben, mithilfe von Parameterdarstellungen Animationen zu generieren.<sup>153</sup>

*Bewegungen*<sup>154</sup> von Körpern im Raum lassen sich durch die Angabe von Parameterdarstellungen für die Koordinaten der die Objekte bestimmenden Punkte beschreiben. Recht einfach ist es beispielsweise, eine Kugel auf einer Bahn, d. h. einer Kurve im Raum, zu bewegen. Bei geradlinigen Bewegungsbahnen ist dazu `clock` als Parameter in die Parameterdarstellungen entsprechender Geraden einzusetzen (Beispielvideo: **Animation1-Strecke.mpg**). Eine Bewegung einer Kugel auf einem Kreis (z. B. dem Kreis

<sup>151</sup>Theoretisch wäre es natürlich möglich, jedes Bild einzeln zu generieren, aufgrund der hohen Zahl benötigter Bilder ist jedoch ein derartiges Vorgehen nicht in Betracht zu ziehen.

<sup>152</sup>Um mit POV-Ray ein Video zu erzeugen, muss neben der Szenenbeschreibung (.pov-Datei) eine Steuer- (.ini-)Datei angelegt werden, in welcher die Anzahl der Bilder und der Definitionsbereich des Parameters `clock` anzugeben sind. Als Ergebnis entstehen nummerierte Einzelbilder, die ein externes Programm in ein Video umwandeln kann. Auf die technischen Einzelheiten soll hier nicht näher eingegangen werden, sie sind in einer Kurzanleitung in Anhang A auf S. 339f. beschrieben.

<sup>153</sup>Für alle aufgeführten Beispiele befinden sich Videos (.mpg-Dateien) im Ordner 2-6 (siehe S. 365).

<sup>154</sup>Der Begriff der Bewegung wird in diesem Abschnitt nicht im „mathematischen“ Sinne einer Abbildung, sondern im physikalischen Sinne der kontinuierlichen Positionsänderung von Objekten gebraucht.

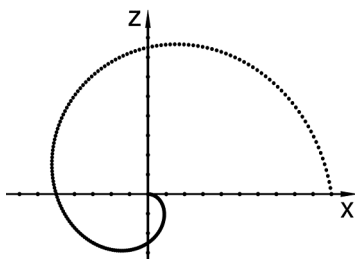
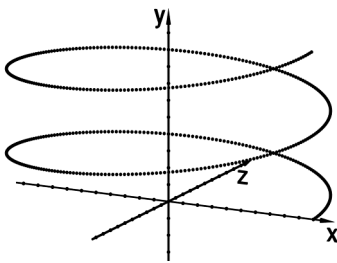
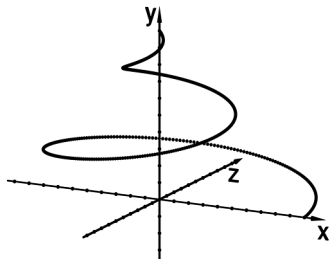
in der  $x$ - $z$ -Ebene mit dem Radius 10 und dem Mittelpunkt im Koordinatenursprung) kann in analoger Weise durch

$$\begin{aligned}x_m(t) &= 10 \cdot \cos(2\pi \cdot t) \\y_m(t) &= 0 \\z_m(t) &= 10 \cdot \sin(2\pi \cdot t)\end{aligned} \quad (t \in [0; 1], t = \text{clock})^{155}$$

beschrieben und in POV-Ray durch

```
sphere <10*cos(2*pi*clock), 0, 10*sin(2*pi*clock) >
```

generiert werden (Video: **Animation2-Kreis.mpg**). Weitere Kurven, die als Bahnkurven genutzt werden können, sind u. a. archimedische Spiralen in zu Koordinatenebenen parallelen Ebenen, Schraubenlinien sowie konische Spiralen. Die folgenden Parameterdarstellungen (die sich durch Variationen der Parameterdarstellung des Kreises ergeben) beschreiben Beispiele derartiger Kurven.<sup>156</sup>

Archimedische Spirale	Schraubenlinie (Helix)	Konische Spirale
$x_m(t) = 10 \cdot (1-t) \cdot \cos(2\pi t)$ $y_m(t) = 0$ $z_m(t) = 10 \cdot (1-t) \cdot \sin(2\pi t)$	$x_m(t) = 10 \cdot \cos(4\pi t)$ $y_m(t) = 10 \cdot t$ $z_m(t) = 10 \cdot \sin(4\pi t)$	$x_m(t) = 10 \cdot (1-t) \cdot \cos(4\pi t)$ $y_m(t) = 10 \cdot t$ $z_m(t) = 10 \cdot (1-t) \cdot \sin(4\pi t)$
		

Durch die animierte Darstellung von Bewegungsbahnen können einfache *physikalische Simulationen* durchgeführt werden. So lässt sich z. B. der schräge Wurf vektoriell durch

$$\vec{x} = \vec{x}_0 + \vec{v}_0 \cdot t + \frac{1}{2} \vec{g} \cdot t^2$$

<sup>155</sup>Als Definitionsintervall für den Parameter `clock` wurde für alle Beispiele  $[0; 1]$  gewählt. Eine solche Normierung ist im Sinne der Übersichtlichkeit vor allem dann sinnvoll, wenn mehrere Objekte bzw. Eigenschaften synchron animiert werden sollen.

<sup>156</sup>Die zugehörigen Beispieldemos sind **Animation3-Spirale.mpg**, **Animation4-Schraubenlinie.mpg** und **Animation5-Kon-Spirale.mpg**. Auf einen Abdruck der POV-Ray-Anweisungen wurde verzichtet, sie ergeben sich aus den Parameterdarstellungen wie bei dem Beispiel des Kreises. Die POV-Ray-Szenenbeschreibungen aller Beispiele befinden sich mit den Videos in dem Ordner 2-6 (siehe S. 365). Die Abbildungen stellen 200 bzw. 500 Positionen der Kugel während der Bewegung entlang der Kurven mit jeweils gleichen Zeitabständen dar. Für ihre Erzeugung wurde anstelle des Zeitparameters `clock` eine Laufvariable in einer Zählschleife eingesetzt, z. B. bei der Schraubenlinie:

```
#declare schritt = 0;
#while (schritt < 500)
#declare t = schritt/500;
sphere { <10*cos(4*pi*t) , 10*t , 10*sin(4*pi*t)> 0.1 }
#declare schritt = schritt + 1;
#end
```

beschreiben, wobei  $\vec{x}_0$  der Ortsvektor des Abwurfpunktes,  $\vec{v}_0$  der Vektor der Abwurfgeschwindigkeit und  $\vec{g}$  der Fallbeschleunigungsvektor ist. So entstand durch die Szenenbeschreibung

```
#declare v0 = <5,5,0>;          // Anfangsgeschwindigkeit
#declare g = <0,-10,0> ;        // Fallbeschleunigung
sphere { <-2.5,0,0> + v0*clock + g/2*clock*clock , 0.25}
```

das Video **Schraeger-Wurf.mpg**, welches die Bewegung einer Kugel entlang der durch die o. a. Gleichung beschriebenen Wurfparabel zeigt.

Durch Parameterdarstellungen beschriebene Bewegungsbahnen können nicht nur für die Animation geometrischer Objekte, sondern auch zur Erzeugung von „*Kamerafahrten*“ bzw. „*flügen*“ genutzt werden. Dazu ist die Position der Kamera durch die Parameterdarstellung einer Kurve auszudrücken, z. B. (wenn sich die Kamera auf einer archimedischen Spirale in einer Ebene parallel zur  $x$ - $z$ -Ebene bewegen soll) durch:<sup>157</sup>

```
#declare r = 60 - 40*clock;
camera { location <r*cos(2*pi*clock), 8, r*sin(2*pi*clock)>
          angle 12 look_at <0,0,0> }
```

Häufig werden neben geometrischen Körpern und Kameras auch die Zielpunkte von Scheinwerfern (Lichtquellen, die nur einen kegelförmigen Ausschnitt des Raumes beleuchten) auf Bahnen bewegt.<sup>158</sup>

## 2.6.2 Zeitabhängige Beschreibung von Transformationen

Bewegungsanimationen können nicht nur durch die zeitabhängige Beschreibung von Koordinaten, sondern auch durch die Darstellung der *Bestimmungsgrößen von Transformationen* (z. B. Verschiebungsvektoren, Drehwinkel, Streckfaktoren) in Abhängigkeit von der Zeit erzeugt werden. Als einfaches Beispiel zeigt das Video **Transformation-Torus1.mpg** die Transformation des in Abb. 2.47 auf S. 113 links dargestellten Torus in den dort rechts abgebildeten Torus. Es entstand durch die Anweisung

```
scale (1+clock*0.5) rotate clock*<60,30,0> translate clock*<-0.5,0.5,0.5> .
```

Dadurch werden die Skalierungsfaktoren, die Drehwinkel und der Verschiebungsvektor gleichzeitig vergrößert. Soll der Torus jedoch z. B. zuerst skaliert, dann um die  $x$ -Achse, danach um die  $y$ -Achse gedreht und schließlich verschoben werden, so ist eine Aufteilung in Parameterintervalle nötig. Im folgenden Beispiel wurde für **clock** das Intervall  $[0; 4]$  gesetzt und durch folgende Fallunterscheidungen eine zeitlich voneinander getrennte Ausführung der Transformationen erreicht (Video **Transformation-Torus2.mpg**):<sup>159</sup>

<sup>157</sup>Beispielvideos für Flüge einer Kamera auf einem Kreis, einer Spirale sowie auf einer konischen Spirale sind **Kameraflug1.mpg** – **Kameraflug3.mpg** in dem Ordner 2-6.

<sup>158</sup>Als Beispiel zeigt das Video **Scheinwerferanimation.mpg** die Bewegung dreier Lichtkegel mit verschiedenen Lichtfarben und Zielpunkten, die sich auf drei (jeweils um  $120^\circ$  gegeneinander versetzten) Spiralen in einer Ebene bewegen. An der Überlagerung der drei Lichtkegel am Ende der Animation wird die Addition der Grundfarben Rot, Grün und Blau zu Weiß sichtbar (siehe Abschnitt 2.1.2).

<sup>159</sup>Da die **#if**-Anweisung in POV-Ray keine alternativen Bedingungen (in der Art von **elseif**) zulässt, musste die Fallunterscheidung durch verschachtelte **#if**-Anweisungen realisiert werden.

```

#if (clock < 1)  scale (1+clock*0.5)
#else
  #if (clock < 2)  scale 1.5  rotate <(clock-1)*60,0,0>
  #else
    #if (clock < 3)  scale 1.5  rotate <60,(clock-2)*30,0>
    #else  scale 1.5  rotate <60,30,0>  translate (clock-3)*<-0.5,0.5,0.5>
    #end
  #end
#end
#end

```

Wie bereits in Abschnitt 2.4.1 erwähnt wurde, ist die Reihenfolge, in der Transformationen ausgeführt werden, von Bedeutung. Besonders deutlich wird dies bei der Animation der Erd- und Mondrotation (Video: *Erde+Mond-Rotation.mpg*):<sup>160</sup>

```

sphere { <0,0,0>, 12.75  rotate <0, -360*clock*27, 0> }      // Erde
sphere { <0,0,0>, 3.48  rotate <0, -360*clock, 0>             // Mond
        translate <30, 0, 0>
        rotate <0, -360*clock, 0> } .

```

Beide Drehungen des Mondes erfolgen um die  $y$ -Achse. Da der Mittelpunkt des Mondes zunächst im Koordinatenursprung liegt, ist die erste Drehung eine Drehung um sich selbst. Nach der Verschiebung ist der Mondmittelpunkt nicht mehr der Koordinatenursprung; die zweite Drehung beschreibt die Drehung des Mondes um die Erde.

Ein weiteres Beispiel für die zeitabhängige Beschreibung geometrischer Transformationen ist die Simulation einer auf einer Ebene rollenden Kugel (im einfachsten Falle auf einer geradlinigen Bewegungsbahn, siehe Beispielvideo *Rollende-Kugel.mpg*). Dazu wird die Kugel mit Mittelpunkt im Koordinatenursprung erzeugt und um eine zu der Ebene parallele, durch den Mittelpunkt der Kugel verlaufende und zur Bewegungsrichtung der Kugel senkrechte Achse zeitabhängig gedreht. Anschließend erfolgt eine zeitabhängige Verschiebung der Kugel:

```

sphere{<0,0,0> 0.5  rotate <0,0,-6/pi*clock*360>  translate <-3+6*clock,0,0>}.

```

Dabei sind die Winkelgeschwindigkeit der Drehung und die Translationsgeschwindigkeit so zu synchronisieren, dass der in einer Zeiteinheit durch die Verschiebung zurückgelegte Weg gleich der Bogenlänge des durch den Drehwinkel beschriebenen Großkreisbogens ist. In dem dargestellten Beispiel bewegt sich die Kugel durch Translation in einer Zeiteinheit um 6 Längeneinheiten und dreht sich um den Winkel  $\varphi = -\frac{6}{\pi} \cdot 360^\circ$ . Dem entspricht eine Bogenlänge von  $\frac{\varphi}{360^\circ} \cdot 2\pi r = -\frac{6}{\pi} \cdot 2\pi r = -6$ , da der Radius der Kugel 0,5 beträgt. Die unterschiedlichen Vorzeichen sind notwendig, da sich Punkte der Kugeloberfläche in den Zeitpunkten, in denen sie beim Rollen den „Boden“ berühren, aus Sicht der Kugel entgegengesetzt zur Rollrichtung bewegen.

<sup>160</sup>Eine Drehung des Mondes um die Erde und eine Drehung um seine eigene Achse dauern gleich lange (ca. 27 Tage und 8 Stunden). Die Größenverhältnisse von Erde und Mond sowie die Umlaufzeiten wurden bei der Erstellung der Animation berücksichtigt. Der Abstand des Mondes zur Erde konnte jedoch nicht maßstabsgetreu übernommen werden, da ansonsten Erde und Mond viel zu klein im Video dargestellt worden wären.



### 2.6.3 Animierte Formen, Farben und Oberflächeneigenschaften

Neben Bewegungen lassen sich durch zeitabhängig beschriebene affine Abbildungen auch Verformungen (gleichmäßige und ungleichmäßige Streckungen sowie Scherungen, siehe Abschnitt 2.4.1) von Körpern animieren. Weiter gehende animierte Formänderungen sind durch zeitabhängige Darstellungen von Konstanten oder Parameterintervallen bei Parameterdarstellungen von Flächen möglich. Werden z. B. bei einer durch eine Parameterdarstellung der Form

$$x(u, v) = x_m + r \cdot \cos(u) \cdot \cos(v), \quad y(u, v) = y_m + r \cdot \sin(v), \quad z(u, v) = z_m + r \cdot \sin(u) \cdot \cos(v)$$

beschriebenen Kugel die Intervalle der Parameter  $u$  und  $v$  mittels

$$u \in (-\pi; -\pi + 2\pi t], \quad v \in [-\frac{\pi}{2}; -\frac{\pi}{2} + \pi t] \quad (t \in [0; 1], \quad t = \text{clock})$$

animiert, so entsteht ein Video, bei dem sich die Kugelfläche kontinuierlich „aufbaut“.<sup>161</sup>

Eine weitere durch zeitabhängige Beschreibung von Parameterintervallen generierte Animation zeigt das „Wachsen“ einer DINI-Fläche<sup>162</sup> mit der Parameterdarstellung

$$x(u, v) = a \cdot \cos u \cdot \sin v, \quad y(u, v) = \cos v + \ln\left(\tan \frac{v}{2}\right) + b \cdot u, \quad z(u, v) = a \cdot \sin u \cdot \sin v$$

und den animierten Parameterintervallen

$$u \in [0; 4\pi t], \quad v \in [0, 0005; 2t] \quad (t \in [0; 1], \quad t = \text{clock}).^{163}$$

Neben der Lage und geometrischen Eigenschaften von Objekten können auch viele andere Objekt- und Szeneneigenschaften wie Farben und Reflexionsvermögen animiert werden; als Beispiel zeigt das Video `Farbanimation.mpg` eine Kugel, die mit fortschreitender Zeit stärker reflektiert und deren Farbe sich dabei von gelb nach blau verändert:

$$\text{color rgb} < 1 - \text{clock}, 1 - \text{clock}, \text{clock} > .^{164}$$

<sup>161</sup>Siehe das Video `Kugel-Animation-Parintervalle.mpg`; Abb. 2.85 zeigt eine „Momentaufnahme“.

<sup>162</sup>Eine Dini-Fläche ist eine Fläche konstanter negativer Krümmung, benannt nach ULISSE DINI (1845–1918). Abb. 2.86 zeigt eine Momentaufnahme aus dem Video `Diniflaeche.mpg` der animierten Dini-Fläche. Die Konstanten in der Parameterdarstellung wurden  $a = 1$  und  $b = 0,2$  gesetzt.

<sup>163</sup>Der Parameter  $v$  darf nicht bei 0 beginnen, da dann  $\tan \frac{v}{2} = 0$  wäre und  $\ln 0$  nicht definiert ist.

<sup>164</sup>Zur Beschreibung von Farben durch RGB-Werte siehe Abschnitt 2.1.2. Neben der Farbe der Kugel wurden die Reflexionskoeffizienten der Kugel und der Ebene sowie die Intensität und Größe der Glanzpunkte auf der Kugel (`phong` und `phong_size`, siehe Abschnitt 2.5.2) zeitabhängig verändert.

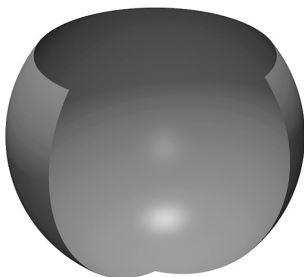


Abbildung 2.85: Teil einer Sphäre

Abbildung 2.86: Dini-Fläche

### 2.6.4 Keyframe- und Pfadanimationen, komplexere Techniken

Für die Erzeugung aufwändiger, realitätsnaher Animationen bis hin zu Bewegungen von Lebewesen ist die explizite, von einem Zeitparameter abhängige Beschreibung der zu animierenden Größen praktisch nicht möglich. 3D-Grafikprogramme enthalten daher verschiedene Hilfsmittel, um Animationen zu erzeugen. So können Bewegungsbahnen (Pfade) durch das Setzen von Stützpunkten und die Beeinflussung von Tangenten mit der Maus „gezeichnet“ werden; die Software generiert daraus Bézier- oder interpolierende Splineskurven (Abschnitt 2.3.6), entlang derer sich Objekte bewegen.

Die klassische Vorgehensweise zur Erzeugung von Animationen ist die *Keyframe-Technik*. Für eine gewisse Zahl von Einzelbildern (Schlüsselszenen bzw. Keyframes) werden Objekte meist mausgesteuert an die gewünschten Positionen gebracht, gedreht, eventuell verformt sowie durch die Eingabe beliebiger Werte in ihren geometrischen oder optischen Eigenschaften verändert. Keyframe-Animationsmodule berechnen die zwischen den Keyframes liegenden Bilder von Animationen,<sup>165</sup> indem sie die für die Keyframes festgelegten Eigenschaften interpolieren. Für die Interpolation kommen meist kubische Splines zur Anwendung. Mitunter werden auch stückweise lineare Interpolationen oder Interpolationen durch andere Klassen von Funktionen vorgenommen.

Keyframe-Techniken und Manipulationen von Bewegungspfaden werden häufig miteinander kombiniert. Auf diese Weise entstand das mit der Software 3ds max angefertigte Video *Kegelschnitte.mpg*, das den Übergang der verschiedenen Kegelschnitte ineinander bei Drehung der Schnittebene zeigt. Während diese Drehung leicht beschrieben werden kann, ist die Animation der Kamera mit Richtungs- und Geschwindigkeitswechseln, welche die Schnittkurven deutlich sichtbar werden lassen, komplizierter und wurde deshalb mithilfe von Keyframes und eines Bewegungspfades erzeugt.<sup>166</sup>

<sup>165</sup>Die Berechnung der „Zwischenbilder“ wird als *Tweening* (von Inbetweening) bezeichnet. Die Keyframe-Technik stammt aus dem klassischen Zeichentrickfilm, bei dem einige Zeichner Schlüsselszenen zeichnen und die Zwischenbilder von anderen (schlechter bezahlten) Zeichnern angefertigt werden.

<sup>166</sup> Abb. 2.87 a) zeigt die Funktionsgraphen der drei Koordinaten der Kamera; die Keyframes sind durch kleine Quadrate gekennzeichnet. Eine Darstellung der Keyframes als Miniaturen ermöglicht das „Storyboard“ der Software Carrara; in Abb. 2.87 b) ist ein Ausschnitt daraus dargestellt.

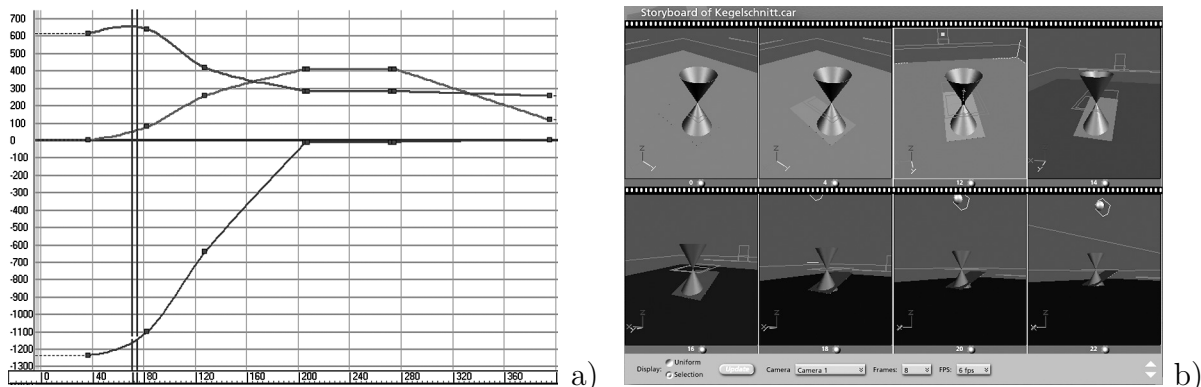


Abbildung 2.87: a) Spuransicht; b) Storyboard bei Keyframe-Animationen

Da sehr komplexe Computeranimationen z. B. in der Filmproduktion erheblich an Bedeutung gewonnen haben und durch die Steigerung der Rechenleistung von Computern praktisch realisierbar wurden, hielten in den vergangenen ca. 10 Jahren hoch entwickelte Animationstechnologien für spezielle Anwendungsgebiete Einzug in die führenden kommerziellen 3D-Softwarepakete. So kommen u. a. für die Animation von Figuren (Charakter Animation) *gegliederte Strukturen* wie Skelette zum Einsatz, mithilfe derer *hierarchische Bewegungen* simuliert werden können. Auf diese Weise lassen sich Bewegungsabläufe wie der menschliche Gang, bei dem Bewegungen von Gliedmaßen in hierarchischer Weise Bewegungen anderer Gliedmaßen nach sich ziehen, simulieren.<sup>167</sup> Durch Verfahren der *inversen Kinematik* können zudem abhängige Glieder von Bewegungshierarchien die Bewegung der übergeordneten Glieder beeinflussen – die Bewegung eines Fingers zieht so z. B. eine Bewegung des entsprechenden Ellenbogens nach sich. Mithilfe von Techniken der *Freiformdeformation* (FFD) ist es möglich, als Polygonnetze repräsentierte Muskeln und Hautoberflächen in Abhängigkeit von den Steuerpunkten gegliederter Strukturen (Skelette) zu animieren.<sup>168</sup>

Neben der Animation der Bewegungsabläufe von Lebewesen haben auch *Simulationen komplexer physikalischer Sachverhalte* eine hohe Bedeutung bei der Entwicklung von Software für Computeranimationen. Dazu gehört die Simulation von Verformungen „weicher“ Körper bei der Einwirkung von Kräften (Soft Body Dynamics). Für die Animation sehr vieler gleichartiger Objekte kommen *Partikelsysteme* zum Einsatz, bei denen das Verhalten einzelner Partikel stochastische Komponenten enthält. Partikelsysteme haben vielfältige Anwendungen, z. B. von der Nachbildung von Fischeschwärmen bis hin zur Simulation fließenden Wassers, gefunden.

Eine tiefer gehende Darstellung der in der professionellen 3D-Computergrafik heute angewendeten Animationstechniken, vor allem in den Bereichen Charakter Animation und Physiksimulation, ginge erheblich über den Rahmen dieser Arbeit hinaus. Es sei daher z. B. auf [177] sowie auf [11], S. 425–494 (unter anderem mit einer ausführlicheren Beschreibung von Keyframe-Techniken) und [257], S. 521–586 verwiesen.

---

<sup>167</sup>Allerdings sind die Steueralgorithmen für Skelette noch nicht ausgereift genug, um Bewegungen von Lebewesen mit überzeugendem Realismus zu simulieren. U. a. aus diesem Grunde wurde für die Animation des Geschöpfes „Gollum“ aus der Filmtrilogie „Herr der Ringe“ ein Schauspieler mit Sensoren versehen, die seine Bewegungsabläufe an wichtigen Steuerpunkten erfassten („Motion Capturing“). Die dabei aufgenommenen Bewegungsdaten konnten dann auf die Steuerpunkte des „Skeletts“ der computergenerierten Figur Gollum übertragen werden, welche nach einer Nachbearbeitung (Komposition des bei der Aufnahme entstandenen Videos mit der Computeranimation) in dem letztendlich produzierten Film den ursprünglichen Schauspieler ersetzte. Für die Herstellung 3D-computergrafisch erzeugter „Trickfilme“ kommen jedoch hauptsächlich mithilfe gegliederter Strukturen erzeugte Computeranimationen zum Einsatz. Den Durchbruch erzielte mit diesen Techniken in der 2. Hälfte der neunziger Jahre die Firma Pixar mit den Filmen „Toy Story“ und „Das große Krabbeln“.

<sup>168</sup>An Verfahren der Animationen von Charakteren wird im Bereich der Computeranimation besonders intensiv gearbeitet. So gibt es z. B. Möglichkeiten, Gesichter virtueller Charaktere in Abhängigkeit von deren Sprache zu animieren. Dies geschieht, indem die Audiodaten, welche die Sprache enthalten, analysiert und über daraus extrahierte Informationen Lippenbewegungen und Gesichtszüge gesteuert werden (siehe [258], S. 500–517).

### 2.6.5 3D-Animationen in Echtzeit, Interaktivität

Alle bisher beschriebenen Animationen mussten zunächst berechnet (gerendert) und als Videodateien gespeichert werden, bevor diese Videos betrachtet werden konnten.<sup>169</sup> Bereits geringe Änderungen machen es notwendig, die Berechnung der Einzelbilder bzw. Videos erneut vorzunehmen. Demgegenüber können *Echtzeit-Animationen* während der Berechnung bereits angeschaut werden – „Rendering in Echtzeit“ bedeutet, dass jedes Einzelbild zu dem Zeitpunkt gerendert wird, zu dem es auf dem Bildschirm erscheint. Für flüssig ablaufende Echtzeit-Animationen muss ein Computer mindestens 25 Bilder in jeder Sekunde berechnen. Dies erfordert die Nutzung schneller Rendering-Verfahren (Scanline Rendering unter Verwendung von Hardware-Beschleunigung, siehe 2.5.4), wobei gegenüber der Produktion von Videos mithilfe hochqualitativer Verfahren wie Raytracing<sup>170</sup> natürlich Abstriche an der Darstellungsqualität hinzunehmen sind.

Von Anwendungsprogrammen oder Computerspielen<sup>171</sup> ausgegebene Echtzeit-3D-Darstellungen werden meist durch Grafikbibliotheken (mit offen gelegten Programmierschnittstellen) erzeugt, die auch auf Grafikkarten vorhandene Grafikprozessoren nutzen können und dadurch eine sehr schnelle Bildberechnung ermöglichen. Die am weitesten verbreiteten dieser Bibliotheken sind OpenGL und Direct3D (siehe Abschnitt 2.7.1). Die meisten 3D-Grafikprogramme nutzen OpenGL für eine schnelle, interaktiv veränderbare Vorschau auf Szenen bei der Konstruktion. Dabei können auch in „Vorschauqualität“ berechnete Animationen in Echtzeit betrachtet werden.

Eine Möglichkeit, 3D-Animationen in Echtzeit unmittelbar durch Parameterdarstellungen von Bewegungskurven zu erzeugen, bietet die objektorientierte Skriptsprache Python in Verbindung mit der Erweiterung VPython (siehe auch Abschnitt 2.7.2). Mithilfe von Python und VPython lässt sich die OpenGL-Schnittstelle unter Verwendung einer vergleichsweise einfachen Syntax ansprechen. Durch den folgenden Quelltext wird die Bewegung einer Kugel entlang einer konischen Spirale animiert:<sup>172</sup>

---

<sup>169</sup>Die Renderzeiten der als Beispiele für die Abschnitte 2.6.1 bis 2.6.4 hergestellten kurzen Videosequenzen betrugen jeweils ca. 5 bis 30 Minuten auf einem 2005/06 gebräuchlichen PC. Ausnahmen bildeten die Animationen `Scheinwerferanimation.mpg` (ca. 3 Stunden) wegen der Nebeneffekte sowie `Kugel-Animation-Parintervalle.mpg` und `Diniflaeche.mpg` (jeweils über 12 Stunden) aufgrund des aufwändigen Verfahrens der Berechnung von Parameterflächen in POV-Ray (vgl. S. 95ff.).

<sup>170</sup>Das Raytracing-Verfahren kommt für Echtzeit-Animationen aus Geschwindigkeitsgründen (zumindest 2005) noch nicht zur Anwendung. Es existieren Projekte (OpenRT Real-Time Ray-Tracing Project: <http://graphics.cs.uni-sb.de/RTRT> und RealStorm: [www.realstorm.com](http://www.realstorm.com)), in denen an der Entwicklung von Echtzeit-Raytracing gearbeitet wird (siehe auch [167]); deren Ergebnisse sind jedoch noch nicht anwendungsreif. In jedem Falle wird innerhalb der nächsten Jahre auch mit Echtzeit-Raytracing nicht die Darstellungsqualität von Software wie POV-Ray, Carrara oder 3ds max möglich sein.

<sup>171</sup>Für moderne Computerspiele ist die Berechnung von 3D-Szenen in Echtzeit unerlässlich, da die hierbei generierten Animationen von der Interaktion mit dem Benutzer abhängig sind – somit ist die Verwendung vorher berechneter Videos nicht möglich.

<sup>172</sup>Aus Platzgründen wird nicht das vollständige Skript, sondern nur der für die Beschreibung der Animation relevante Teil abgedruckt. Das komplette Skript `Kugelbahn-KonischeSpirale.py` befindet sich in dem Ordner 2-6\Echtzeit-Animationen (siehe Anhang D, S. 365).

```

start = vector(-2.5,-2.8,0)           # Startposition
Kugel = sphere (pos=start, color=color.blue, radius=0.2) # Kugel
Spur = curve(color=(0.2,0.2,0.2), radius=0.01)          # Spurkurve
dt = 0.005                                           # Zeitintervall
t = 0.0                                              # Startzeit
while t <= 1:                                       # Beschreibung
    rate(50)                                       # der Animation
    t = t + dt                                    # als Schleife
    Kugel.pos = vector(start.x*(1-t)*cos(4*pi*t),    # Parameter-
                        start.y+2*t,                # darstellung
                        start.x*(1-t)*sin(4*pi*t))
    Spur.append(pos=Kugel.pos)                    # Spurkurve um
                                                    # Punkt ergänzen

```

Durch Ausführen des Skripts mithilfe des Python-Interpreters läuft in Echtzeit eine Animation ab, welche die Bewegung der Kugel zeigt, wobei außerdem die Spur (Bewegungsbahn) der Kugel als Kurve gezeichnet wird. Weitere Python-Skripte, die Animationen auf archimedischen Spiralen und Schraubenlinien sowie einen schrägen Wurf beschreiben, befinden sich in dem Ordner 2-6\Echtzeit-Animationen.<sup>173</sup>

Die Darstellungsqualität mit Python unter Nutzung der OpenGL-Bibliotheken erzeugter Animationen ist deutlich schlechter als z. B. die mit POV-Ray berechneter Videos;<sup>174</sup> auffällig ist u. a. die recht grobe Annäherung von Kugeln durch Polygonnetze.<sup>175</sup> Allerdings ist die Möglichkeit der Darstellung in Echtzeit oft wichtiger als die Qualität.

Die interaktive Veränderung von Parametern mit sofortigem Blick auf die Auswirkungen setzt Echtzeit-Grafikberechnung zwingend voraus. Ein einfaches Beispiel für *interaktive Animationen* beinhaltet das Skript **Schraeger-Wurf-Interaktiv.py** bzw. die ohne Python-Installation ausführbare Version **Schraeger-Wurf-Interaktiv.exe**. Durch Schieberegler können bei dieser Animation die Abwurfgeschwindigkeit sowie der Wurfwinkel eingestellt und die Wirkungen sofort getestet werden.

Echtzeit-Animationen lassen sich auch in einigen Computeralgebrasystemen (CAS, siehe 2.7.5) generieren; auf diese Möglichkeit wird in der vorliegenden Arbeit noch im Zusammenhang mit Unterrichtsvorschlägen eingegangen (siehe Abschnitt 4.3).

<sup>173</sup>Um die Animationen ablaufen zu lassen, ist die Installation der (freien) Softwarepakete Python und VPython erforderlich. Diese sind unter [www.python.org](http://www.python.org) bzw. [www.vpython.org](http://www.vpython.org) erhältlich. Durch Veränderungen von Werten z. B. für Zeitintervalle oder Winkel können die Animationen modifiziert werden. In dem Ordner 2-6\Echtzeit-Animationen-Kompiliert befinden sich außerdem ausführbare Versionen (.exe-Dateien) der Python-Skripte, die unter Windows die Betrachtung der Animationen ermöglichen, ohne Python und VPython zu installieren.

<sup>174</sup>Die Dateien in dem Ordner 2-6 ermöglichen einen direkten Qualitätsvergleich, da dieselben Vorgänge sowohl unter Verwendung von POV-Ray (siehe 2.6.1) als auch von VPython animiert wurden.

<sup>175</sup>Die darüber hinaus bei Echtzeit-3D-Animationen auf vielen Computern auftretende starke „Treppenbildung“ (Aliasing) lässt sich oft durch die Aktivierung von Anti-Aliasing (vgl. 2.5.7) reduzieren. Da die Grafikdarstellung von VPython-Animationen durch die OpenGL-Bibliotheken erfolgt, muss die Einstellung von Anti-Aliasing in den OpenGL-Einstellungen des Grafikkartentreibers erfolgen.

## 2.7 Überblick über Software für die Erstellung 3D-computergrafischer Darstellungen

Es ist im Rahmen der vorliegenden Arbeit nicht möglich, einen detaillierten Überblick über die vielfältigen Möglichkeiten der zahlreichen Anwendungsprogramme für die Erstellung von 3D-Computergrafiken und -animationen zu geben. Andererseits erscheint es im Hinblick auf die im nächsten Kapitel zu treffende Auswahl geeigneter Software für den Unterricht notwendig, auch Programme in die Betrachtungen einzubeziehen, bei denen es sich nicht um 3D-Grafiksoftwarepakete im engeren Sinne handelt, die aber dreidimensionale Grafiken generieren können und bereits im Mathematikunterricht zum Einsatz kommen. Dies betrifft vor allem Computeralgebrasysteme und Software für den Geometrieunterricht. Da dreidimensionale Darstellungen im computerunterstützten Konstruieren eine hohe Bedeutung erlangt haben, erscheint es zudem angebracht, einen Blick auf die diesbezüglichen Vorgehensweisen in CAD-Software zu werfen.

Die genannten Softwarekategorien werden anhand exemplarisch ausgewählter Produkte überblicksartig beschrieben. Damit soll die Grundlage dafür geschaffen werden, eine für die Verwendung im Stoffgebiet Analytische Geometrie geeignete Software auszuwählen.

### 2.7.1 Architektur, Komponenten und Schnittstellen von 3D-Grafiksoftware (Überblick)

Auf die Programmierung von 3D-Grafiksystemen kann in der vorliegenden Arbeit nicht detailliert eingegangen werden, da die Implementierung derartiger Systeme sehr aufwändig und komplex ist.<sup>176</sup> Jedoch ermöglichen es „Schnittstellen“ zwischen den Komponenten von 3D-Grafiksoftware, computergrafische Darstellungen durch recht kurze und übersichtliche Skripte zu generieren, welche hauptsächlich mathematische Beschreibungen der in den zu erstellenden Szenen vorhandenen Objekte beinhalten.

Computergrafische Anwendungsprogramme bestehen aus mehreren Ebenen von der Beschreibung dreidimensionaler Szenen durch den Benutzer bis hin zu ihrer Darstellung auf dem Bildschirm. Abb. 2.88 verdeutlicht den Aufbau von 3D-Grafiksystemen,<sup>177</sup> wo-

---

<sup>176</sup>Prinzipiell können natürlich für alle in diesem Kapitel beschriebenen Modellierungs-, Transformations-, Rendering- und Animationsanwendungen Programme bzw. Module in Programmiersprachen erstellt werden. Beispiele für die Programmierung einfacher Software zur Darstellung dreidimensionaler Objekte als „Drahtgitter“ mithilfe der Sprache Pascal enthalten die (bereits älteren) Veröffentlichungen [23], [59], [60], [62] und [108]. Die Implementierung der wichtigsten Module von 3D-Grafiksoftware unter Verwendung der Programmiersprache C++ wird in [256] beschrieben. Vollständige 3D-Grafiksysteme, die alle Arbeitsschritte von der Modellierung bis hin zur Bildberechnung und evtl. Animation ausführen können, erfordern allerdings jahrelange Entwicklungsarbeiten mehrerer Arbeitsgruppen, die Module für die einzelnen Verarbeitungsschritte (siehe die Übersicht auf S. 84) erstellen.

<sup>177</sup>Das dargestellte Schema verdeutlicht nicht bzw. nur sehr grob, welche Berechnungen in einem 3D-Grafiksystem ausgeführt werden, siehe dazu die Übersicht auf S. 84. Jedoch zeigt Abb. 2.88 die zugänglichen Schnittstellen zwischen den verschiedenen Funktionsgruppen und gibt damit Anhaltspunkte, auf welchen Ebenen die Erstellung bzw. Beschreibung von Grafiken erfolgen kann (vgl. auch [11], S. 3–4).

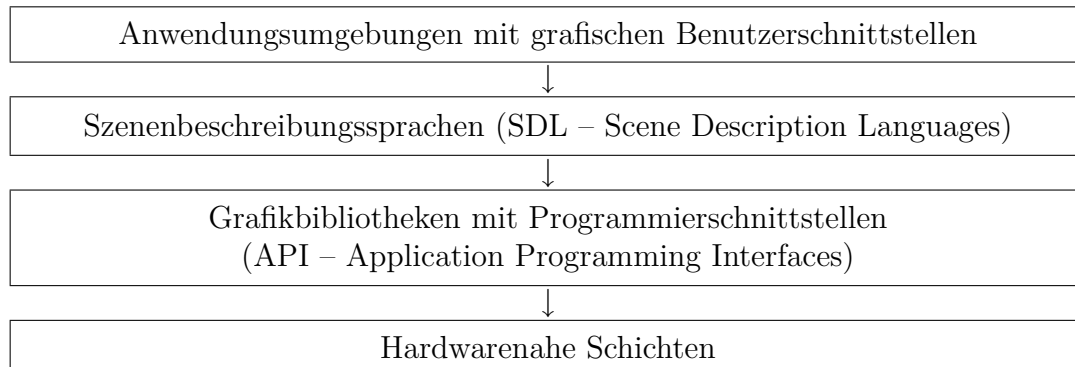


Abbildung 2.88: Schichten und Schnittstellen von 3D-Grafiksystemen

bei jedoch die beiden mittleren Schichten nicht in allen Fällen voneinander zu trennen sind und z. B. bei den meisten Raytracing-Programmen eine Einheit bilden.

Bestandteile der obersten Schicht sind die für Anwender sichtbaren Konstruktionsfenster und Dialogfelder von Grafikprogrammen, in denen sich Objekte mithilfe der Maus erstellen und modifizieren lassen (siehe S. 166). Szenen werden von diesen Programmen in für die jeweilige Software spezifischen Szenendateien gespeichert, die alle durch den Benutzer vorgenommenen Konstruktionen und Einstellungen enthalten – es handelt sich somit um Szenenbeschreibungen, die allerdings meist binär kodiert und nur für die jeweiligen Softwarepakete oder spezielle Konvertierungsprogramme „lesbar“ sind. Jedoch ermöglichen fast alle 3D-Grafikprogramme den Export von Szenen in Beschreibungssprachen wie VRML (Virtual Reality Modeling Language), die 3D-Szenen durch Anweisungen in Textdateien repräsentieren. In vielen Szenenbeschreibungssprachen (auch als „High-Level-API“ bezeichnet) können nicht nur Objektbeschreibungen vorgenommen, sondern auch Programmierstrukturen wie Verzweigungen und Schleifen genutzt werden. Auf einige Szenenbeschreibungssprachen wird im nächsten Abschnitt eingegangen.

Vor der Bildberechnung und -darstellung werden proprietäre oder (z. B. bei VRML) standardisierte Szenenbeschreibungen für die Übergabe an Rendering-Module aufbereitet. Dabei kann es sich um Raytracing-Programme oder um Scanline-Renderer handeln, wobei im letzteren Falle oft die Bibliotheken OpenGL und Direct3D zum Einsatz kommen.<sup>178</sup> Deren Programmierschnittstellen (oft als „Low-Level-API“ bezeichnet) lassen keine unmittelbaren Szenenbeschreibungen mit vertretbarem Aufwand zu und sind für

---

<sup>178</sup>OpenGL (Open Graphics Library) ist für alle Computerplattformen verfügbar; Direct3D existiert nur für die Betriebssysteme von Microsoft und kommt vor allem für die Darstellung von Spielen zum Einsatz. Die OpenGL- und Direct3D-Bibliotheken enthalten Module für die zur Bildberechnung erforderlichen Schritte (siehe S. 84), einschließlich der notwendigen Koordinatentransformationen, Projektionen, Sichtbarkeitsbestimmungen und des Rendering. Die Mehrzahl dieser Verarbeitungsschritte wird auf modernen Computern von der Grafikhardware ausgeführt (siehe Abschnitt 2.5.4), auf deren Funktionen OpenGL bzw. Direct3D zugreifen. Dabei werden ausschließlich Polygonobjekte verarbeitet; OpenGL und Direct3D enthalten Zusatzbibliotheken, die Körper in Polygonnetze umwandeln. Beschreibungen der OpenGL-Routinen und -Schnittstellen enthalten u. a. [189] und [195].

die Kommunikation mit Anwendungsprogrammen konzipiert. Von den bisher in diesem Kapitel anhand von Beispielen betrachteten Softwarepaketen greifen VPython und MuPAD auf OpenGL für die Bildschirmdarstellung von 3D-Grafiken zurück, Carrara und 3ds max können OpenGL für die Vorschau während der Konstruktion nutzen.

Auf der untersten Schicht des Schemas in Abb. 2.88 erfolgt die Übergabe der berechneten Bilder an den Grafikspeicher und von dort an den Monitor. Diese Ebene ist nicht spezifisch für 3D-Darstellungen, sondern realisiert die Ausgabe beliebiger Bilder sowie Videos. Es existieren auch 3D-Grafikprogramme, die Bilder nicht selbst auf dem Bildschirm darstellen, sondern lediglich als Dateien speichern, wobei externe Software die Anzeige auf dem Bildschirm übernimmt.

### 2.7.2 Szenenbeschreibungssprachen und Raytracing-Software

Dreidimensionale Computergrafiken lassen sich durch die Beschreibung in strukturierten und verständlichen Sprachen generieren, die oft als Szenenbeschreibungssprachen (SDL) oder „High-Level-API“ bezeichnet werden (siehe Abb. 2.88). Dazu zählen die Beschreibungssprachen einiger Raytracing-Programme, Beschreibungssprachen wie VRML und entsprechende Module für Programmiersprachen (wie VPython); auch die 3D-Grafikbefehle in CAS können als Szenenbeschreibungssprachen aufgefasst werden.

Um die Syntax verschiedener SDL hinsichtlich der Beschreibung einfacher Objekte miteinander zu vergleichen, wurde eine einfache Szene, die aus einem grauen Zylinder, einem blauen Kegel und einer roten Kugel besteht, mithilfe verschiedener derartiger Sprachen beschrieben und jeweils eine grafische Darstellung erzeugt (siehe Abb. 2.89).<sup>179</sup>

Die Sprache *VRML* (Virtual Reality Modeling Language) entstand 1995 mit der Intention, damit „virtuelle Welten“ innerhalb des World Wide Web zu entwickeln.<sup>180</sup> Sie ermöglicht nicht nur die Erzeugung von 3D-Grafiken und -Animationen, sondern darüber hinaus die Vernetzung von Szenen durch Hyperlinks und die Gestaltung interakti-

<sup>179</sup> Alle generierten Grafiken sowie die zugehörigen Szenenbeschreibungen befinden sich in dem Ordner 2-7 (siehe Anhang D sowie die Hinweise in der Datei `Testszene_Hinweise.txt`).

<sup>180</sup> Allerdings konnten sich VRML und auch die Weiterentwicklung X3D (siehe [www.web3d.org](http://www.web3d.org)) nicht in breitem Umfang als Web-3D-Standard etablieren, was vor allem daran lag, dass es viele konkurrierende Projekte einzelner Hersteller gab, die sich ebenfalls nicht durchsetzen konnten. So existiert bis heute kein verbreiteter Standard für 3D-Grafiken und -animationen im Internet. Dennoch hat VRML eine Bedeutung als offene 3D-Beschreibungssprache und Austauschformat erlangt.

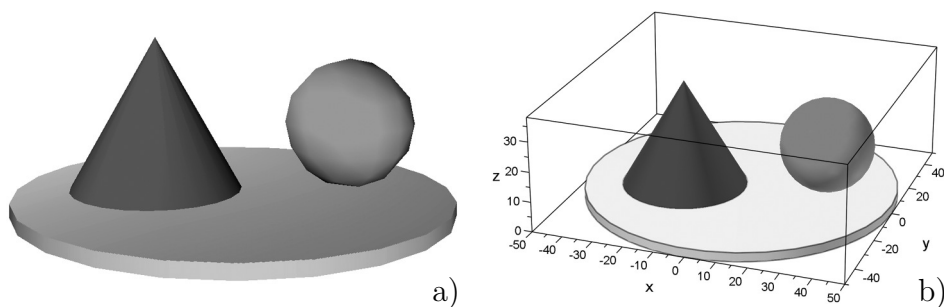


Abbildung 2.89:  
Testszene  
a) VRML  
b) MuPAD



ver Benutzerumgebungen. Diese Elemente erfordern Hierarchien bei der Beschreibung von Objekten, Eigenschaften und Transformationen,<sup>181</sup> was die Beschreibung geometrischer Objekte gegenüber reinen Szenenbeschreibungssprachen etwas erschwert, wie der VRML-Quelltext der in Abb. 2.89 dargestellten Szene zeigt:<sup>182</sup>

```
DEF Zylinder01 Transform { translation 0 1.5 0   children [ Shape {
    appearance Appearance { material Material {diffuseColor 0.6 0.6 0.6}}
    geometry Cylinder { radius 50 height 3 }           } ] }
DEF Kegel01 Transform { translation -17 20.5 -6 children [ Shape {
    appearance Appearance { material Material {diffuseColor 0 0 1} }
    geometry Cone { bottomRadius 20 height 35 }         } ] }
DEF Kugel01 Transform { translation 27 18 5      children [ Shape {
    appearance Appearance { material Material {diffuseColor 1 0 0} }
    geometry Sphere { radius 15 }                     } ] }.
```

Dieselbe Szene kann in der Sprache *VPython* folgendermaßen beschrieben werden:<sup>183</sup>

```
cylinder( pos=(0,0,0), axis=(0,3,0), radius=50, color=(0.6,0.6,0.6) )
cone( pos=(-17,3,-6), axis=(0,38,0), radius=20, color=(0,0,1) )
sphere( pos=(27,18,5), radius=15, color=(1,0,0) ) .
```

Aufgrund der Einbindung der Szenenbeschreibungssprache *VPython* (Visual Python) in die leistungsfähige Programmiersprache Python (siehe u. a. [148] und [176]) lassen sich nicht nur durch einfache Anweisungen dreidimensionale Szenen erzeugen. 3D-Grafiken und -Animationen können unter Nutzung der durch Python gegebenen Programmiermöglichkeiten zu Anwendungsprogrammen erweitert werden, wie bereits das einfache Beispiel der interaktiven Simulation eines schrägen Wurfes (siehe S. 159) zeigt.

Als Szenenbeschreibungssprache können auch die 3D-Grafikbefehle des CAS *MuPAD* aufgefasst werden, welches neben der Visualisierung durch Gleichungen bzw. Parameterdarstellungen gegebener Flächen auch die Darstellung einiger Körper (Würfel, Zylinder, Kegel, Kugeln und Ellipsoide) durch Koordinatenbeschreibungen ermöglicht. Die Testszene (siehe Abb. 2.89) entsteht durch folgende *MuPAD*-Anweisungen:

```
Zylinder := plot::Cylinder(50, [0,0,0], [0,0,3], Color = RGB::Gray)
Kegel := plot::Cone(20, [-17,-6,3], [-17,-6,38], Color = RGB::Blue)
Kugel := plot::Sphere(15, [27, 5, 18], Color = RGB::Red)
plot(Zylinder,Kegel,Kugel).184
```

<sup>181</sup>Diesen Hierarchien liegt die Modellierung von Szenen durch Szenengraphen zugrunde (siehe [105]), was allerdings anhand des hier dargestellten, sehr einfachen Beispiels nicht deutlich wird.

<sup>182</sup>Aus Platzgründen sind, wie auch bei den folgenden Beispielen, nur die Beschreibungen der geometrischen Objekte abgedruckt; die vollständigen Szenen befinden sich in dem Ordner 2-7 (siehe S. 365).

<sup>183</sup>Die gegenüber der VRML-Beschreibung anderen Koordinaten bei der Beschreibung des Kegels und des Zylinders erklären sich dadurch, dass in VRML die Koordinaten der Achsenmittelpunkte und in *VPython* die Koordinaten von Endpunkten der Achsen des Kegels bzw. des Zylinders anzugeben sind.

<sup>184</sup>Da VRML und *VPython* die *y*-Achse als senkrechte Achse darstellen, *MuPAD* jedoch standardmäßig die *z*-Achse, wurden die *y*- und *z*-Koordinaten für das *MuPAD*-Beispiel vertauscht, um die Szene in derselben Weise auf dem Bildschirm darzustellen wie bei den VRML- und *VPython*-Beispieldateien.

Eine Gemeinsamkeit der in VRML, VPython und MuPAD erzeugten Grafiken besteht darin, dass deren Berechnung in Echtzeit erfolgt, wofür von VPython, MuPAD und den meisten VRML-Modulen die OpenGL-Bibliotheken genutzt werden. Dadurch ist die Betrachtung von Szenen aus unterschiedlichen Perspektiven durch Drehungen, Verschiebungen, Vergrößerungen und Verkleinerungen des Bildes mithilfe der Maus möglich.

### Durch Szenenbeschreibungssprachen gesteuerte Raytracing-Software

Wie bereits an mehreren Stellen dieses Kapitels ausgeführt wurde, bestehen in Bezug auf das Verfahren der Bildberechnung, die erreichbare Bildqualität und die Berechnungsdauer erhebliche Unterschiede zwischen Scanline-Rendering-Verfahren, die Echtzeit-Darstellungen ermöglichen, und dem Raytracing-Verfahren, welches sehr hochwertige, fotorealistische Bilder generieren kann, aber keine interaktive Szenenbetrachtung zulässt. Hinsichtlich der Beschreibung dreidimensionaler Szenen existieren jedoch Gemeinsamkeiten zwischen den bisher beschriebenen Szenenbeschreibungssprachen (SDL) und denjenigen Raytracing-Programmen, die durch SDL gesteuert werden. Dies wird durch den Vergleich der für VRML, VPython und MuPAD angegebenen Beschreibungen der Testszene mit der Beschreibung für die Raytracing-Software *POV-Ray* deutlich:

```
cylinder { <0,0,0>, <0,3,0>, 50 pigment { color rgb <0.6,0.6,0.6> } }
cone { <-17,3,-6>, 20, <-17,38,-6>, 0 pigment { color rgb <0,0,1> } }
sphere { <27, 18, 5>, 15 pigment { color rgb <1,0,0> } } .
```

Etwas umfangreichere Beispiele für POV-Ray-Szenenbeschreibungen wurden in diesem Kapitel bereits angegeben. Die POV-Ray-SDL ist eine leistungsfähige Sprache, welche die Modellierung komplexer geometrischer Objekte, wie durch Splinekurven begrenzter Rotationsflächen, Bézier-Patches sowie durch Gleichungen oder Parameterdarstellungen gegebener Flächen, ermöglicht. Sie enthält zudem Anweisungen für die detaillierte Beeinflussung des Erscheinungsbildes von Objektoberflächen.<sup>185</sup> Weiterhin sind Programmierstrukturen wie Verzweigungen und Schleifen Bestandteile der POV-Ray-SDL.<sup>186</sup>

POV-Ray wird seit 1987 als freie Software mit offenen Quellen von einer großen Gruppe von Programmierern weltweit beständig weiter entwickelt und steht für Windows-, Macintosh- und Linux-Betriebssysteme zur Verfügung.<sup>187</sup> Die Verfügbarkeit der Quelltexte hat zudem dazu geführt, dass eine Reihe von Modifikationen, ergänzenden Zu-

<sup>185</sup>In dem hier dargestellten Beispiel wurden den Objekten im Sinne der Vergleichbarkeit mit den VRML-, VPython- und MuPAD-Szenenbeschreibungen nur Farben zugeordnet. Abschnitt 2.5 enthält jedoch u. a. Beispiele für die Beeinflussung von Mattheit, Glanzpunkten, Reflexion, Transparenz und Brechung in POV-Ray. Die Abbildungen in Anhang B, die zum großen Teil mithilfe von POV-Ray erstellt wurden, vermitteln einen Eindruck von der erreichbaren Bildqualität.

<sup>186</sup>Siehe dazu die Beispiele auf S. 153–154 und in Fußnote 156 auf S. 152. Detailliertere Beschreibungen der POV-Ray-SDL und eine Reihe weiterer Beispiele finden sich im Handbuch zu dieser Software ([200]) sowie u. a. in [54], [147] und [267].

<sup>187</sup>Andere freie Raytracing-Programme sind u. a. Polyray, dessen Entwicklung jedoch eingestellt wurde, und das relativ junge Yafray (siehe [www.yafray.org](http://www.yafray.org)). Allerdings hat kein anderes freies Programm auch nur annähernd den Reifegrad von POV-Ray erreicht.

satzprogrammen sowie Konvertierungsmodulen entstanden. Letztere ermöglichen die Bearbeitung in anderen Programmen erstellter Szenen mithilfe von POV-Ray.<sup>188</sup>

Im kommerziellen Bereich haben die SDL-gesteuerten Raytracing-Systeme *Mental Ray* und *Pixar RenderMan* eine weite Verbreitung erreicht.<sup>189</sup> RenderMan, von der Firma Pixar vor allem für das Rendering der eigenen Filmproduktionen entwickelt,<sup>190</sup> ermöglicht eine besonders effiziente Bildberechnung, was bei aufwändigen Spielfilmproduktionen zu den Hauptkriterien für die Anwendbarkeit einer Software gehört. Die Berliner Firma Mental Images GmbH verfolgt bei der Entwicklung von Mental Ray vor allem das Ziel, durch die Integration fortgeschrittener Renderalgorithmen ein Höchstmaß an Fotorealismus zu erreichen.<sup>191</sup>

In Bezug auf die grafischen Möglichkeiten gehen die (recht teuren) Systeme RenderMan und Mental Ray nicht wesentlich über den von POV-Ray inzwischen ebenfalls erreichten Stand hinaus. Erhebliche Unterschiede bestehen jedoch im Hinblick auf die Tauglichkeit für die Produktion großer Projekte. Dazu bieten sowohl RenderMan als auch Mental Ray ausgereifte Möglichkeiten, Rendereaufgaben auf große Netzwerke mit mehreren Hundert Computern zu verteilen, was z. B. für die Produktion von Spielfilmen unerlässlich ist.

Sowohl RenderMan als auch Mental Ray verfügen über leistungsfähige Szenenbeschreibungssprachen,<sup>192</sup> die jedoch nicht vorrangig für die direkte Verwendung durch Eingabe von Textanweisungen konzipiert sind (wenngleich die Möglichkeit dazu besteht), sondern vorrangig der Datenübernahme aus Programmen mit grafischen Benutzeroberflächen dienen (u. a. Softimage, Maya und 3ds max, siehe nächster Abschnitt). Eine möglichst umfassende Kompatibilität zu diesen Softwarepaketen erfordert in stärkerem Maße hierarchische Gliederungen der SDL, was zur Folge hat, dass die direkte Beschreibung von Szenen etwas aufwändiger ist als z. B. in POV-Ray. Die SDL von Mental Ray beinhaltet zudem keine Befehle für geometrische Grundkörper, kann jedoch Polygonnetze, Bézier- und B-Splineflächen (einschließlich NURBS) direkt verarbeiten.

---

<sup>188</sup>Informationen über die zahlreichen verfügbaren Erweiterungen geben die POV-Ray-Homepage [www.povray.org](http://www.povray.org) sowie die POV-Ray-Wiki [www.wikipov.org](http://www.wikipov.org).

<sup>189</sup>Siehe [www.mentalimages.com](http://www.mentalimages.com) sowie <https://renderman.pixar.com>.

<sup>190</sup>Pixar produzierte 1995 den ersten vollständig computergenerierten Spielfilm „Toy Story“ und war auch mit den seither entstandenen Filmen bis hin zu „The Incredibles“ (Ende 2004) sehr erfolgreich. Die von Pixar produzierten Filme geben exemplarisch Auskunft über die Fortschritte der 3D-Computergrafik und -animation in den vergangenen zehn Jahren. Innovationen wie etwa organische Modellierung und Animation (in „Das große Krabbeln“, 1998), Simulationen von Haaren („Monster AG“, 2001) und Unterwassersimulationen („Findet Nemo“, 2003) wurden hier in ausgereiften Großproduktionen praktisch umgesetzt. Dazu trieb Pixar die Entwicklung von RenderMan beständig voran.

<sup>191</sup>Wie bereits in Abschnitt 2.5.6 erwähnt wurde, war Mental Ray das erste System, welches das Photon-Tracing-Verfahren für die Simulation von globaler Illumination und von Kaustiken anwendete.

<sup>192</sup>Die SDL von RenderMan ist u. a. in [240] beschrieben, die von Mental Ray in [49].

### 2.7.3 3D-Grafiksoftware mit grafischen Benutzeroberflächen

Während bei durch Szenenbeschreibungssprachen gesteuerten Grafikprogrammen alle Objekte und Eigenschaften durch Textanweisungen beschrieben werden müssen, kommen in der praktischen Anwendung vor allem Programme zum Einsatz, welche die Erstellung und Positionierung geometrischer Objekte durch Klicken und Ziehen mit der Maus und die Festlegung von Eigenschaften in Dialogfeldern zulassen. Abb. 2.90 zeigt die Benutzeroberfläche der Software *eovia Carrara 4.1 Pro*;<sup>193</sup> die Oberflächen der meisten 3D-Grafiksoftwarepakete sind vergleichbar aufgebaut.<sup>194</sup> Im Folgenden werden die wichtigsten Bestandteile der Benutzeroberfläche anhand der Kennzeichnungen (Ziffern) in Abb. 2.90 kurz beschrieben.

- (1) Zentraler Bestandteil jeder 3D-Grafiksoftware mit grafischer Benutzeroberfläche ist ein *Konstruktionsfenster*, in dem Objekte platziert, verschoben, gedreht und skaliert werden. Es lassen sich (wie in Abb. 2.90) mehrere Ansichten oder auch nur einzelne Ansichten, dann entsprechend größer, darstellen. In den Ansichtsfenstern sind sowohl Drahtgitter- als auch schattierte Darstellungsmodi wählbar.

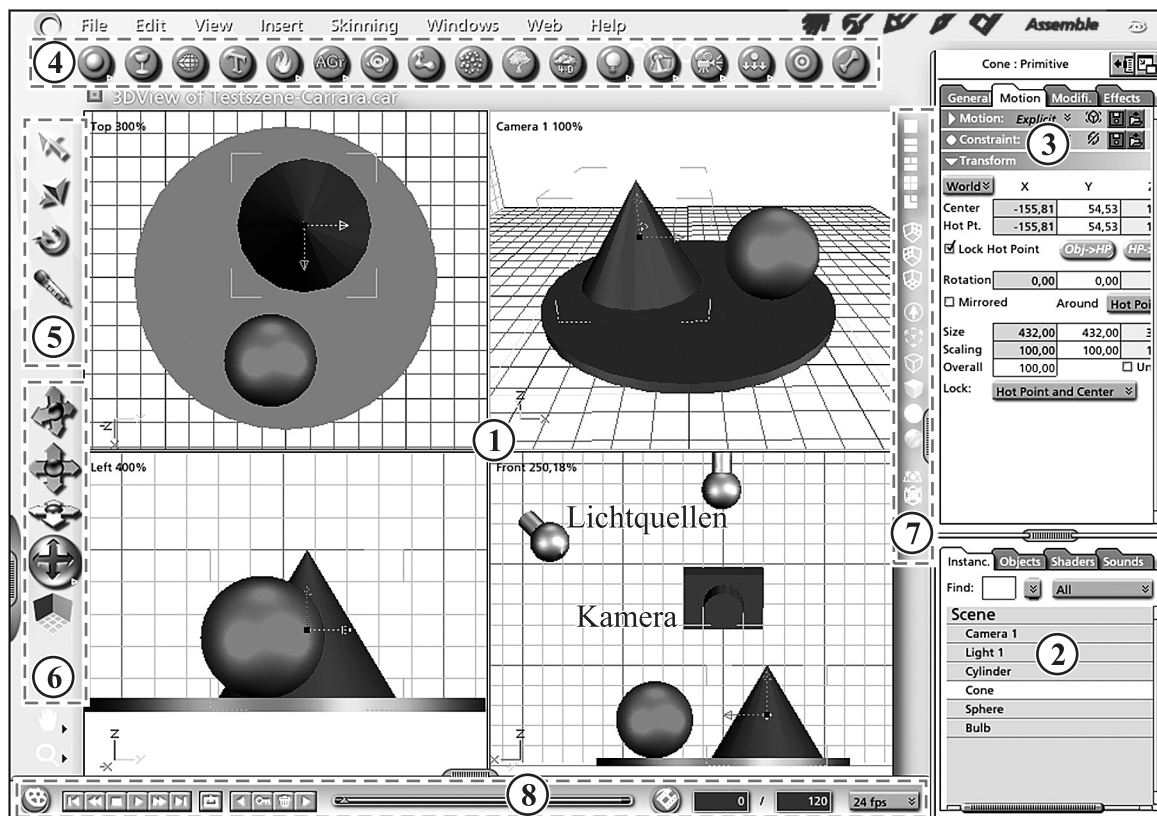


Abbildung 2.90: Benutzeroberfläche der Software Carrara 4.1 Pro

<sup>193</sup>Weitere Informationen zu der Software Carrara finden sich in [45], [63] sowie unter [www.daz3d.com](http://www.daz3d.com). Dort kann auch eine 30 Tage nutzbare Testversion dieser Software heruntergeladen werden.

<sup>194</sup>Siehe z. B. das in Abb. 2.17 auf S. 82 dargestellte Konstruktionsfenster der Software 3ds max.

Seit einigen Jahren ist es in vielen Softwarepaketen nicht mehr unabdingbar, mit mehreren Ansichten zu arbeiten, um Objekte parallel zu Koordinatenachsen zu verschieben, sie um zu Koordinatenachsen parallele Geraden zu drehen oder in ausgewählten Koordinatenrichtungen zu skalieren, da dies nun auch „Anfasser“ in perspektivischen Ansichten ermöglichen.

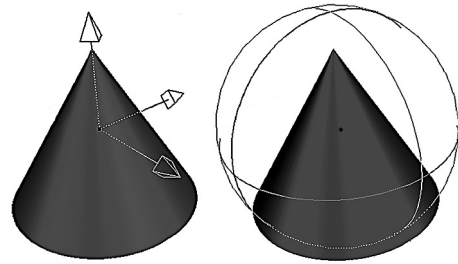


Abbildung 2.91: „Anfasser“ zum Verschieben und Drehen von Objekten

- (2) *Auswahlfenster* geben einen Überblick über alle in einer Szene enthaltenen Objekte, Texturen und anderen Ressourcen. Die Auswahl von Objekten für die Bearbeitung ist sowohl in Konstruktions- als auch in Auswahlfenstern möglich.
- (3) Zusätzlich zur Positionierung und Transformation von Objekten mithilfe der Maus ermöglichen viele Softwarepakete auch die numerische Eingabe von Koordinaten, Abmessungen und Drehwinkeln. Im *Eigenschaftenfenster* werden ausgewählten Objekten zudem Attribute wie Schattenwurf, Textur sowie Effekte zugewiesen.
- (4) Durch Auswahl von Konstruktionswerkzeugen in *Werkzeugleisten* oder Menüs und anschließendes Klicken und Ziehen im Konstruktionsfenster erfolgt die Erstellung von Objekten. Die Werkzeugleiste (4) zeigt, dass Carrara vielfältige Objekttypen erzeugen kann, u. a. Grundkörper, Rotationskörper, Spline-Flächen, Polygonnetze, durch Parameterdarstellungen beschriebene Flächen, Terrains, Partikelsysteme, Bäume sowie verschiedene Arten von Lichtquellen und Kameras.
- (5) *Transformationswerkzeuge* werden genutzt, um Objekte zu verschieben, zu skalieren und zu drehen. Die Pipette überträgt Texturen zwischen Objekten.
- (6) *Navigationswerkzeuge* dienen der Wahl des darzustellenden Ausschnitts der Szene und der Perspektive. Damit erfolgt die Festlegung der Kameraeinstellungen.
- (7) Mithilfe der *Ansichtsleiste* werden die im Konstruktionsfenster darzustellenden Ansichten sowie die Arten der Darstellung gewählt.
- (8) Die *Zeitleiste* dient der Navigation in Animationen durch die Auswahl zu bearbeitender Frames. Durch Vergrößern (Aufklappen) lassen sich Keyframes (siehe Abschnitt 2.6.4) anzeigen und zeitlich verschieben.

Neben diesen Grundfunktionen zur Szenenbearbeitung existieren weitere Aufgabenbereiche für die Modellierung komplexer geometrischer Formen, die Beschreibung von Texturen (Oberflächenerscheinungen), die Bildberechnung (Rendering) sowie für die Bearbeitung von Animationen. Die Benutzeroberfläche von Carrara ist hinsichtlich dieser Aufgabenbereiche in so genannte Räume unterteilt, die sich über die am oberen Rand des Programmfensters rechts angeordneten Symbole auswählen lassen:

- *Assemble Room* (das in Abb. 2.90 dargestellte Hauptfenster),
- *Model Room* zur Bearbeitung von Spline-Objekten (siehe S. 105), Polygonobjekten (Vertex Modeling, S. 112), Terrains und weiteren geometrischen Formen;

- *Texture Room* für die Zuweisung von Farben und Mustern, Transparenz, Reflexions- und Brechungseigenschaften (siehe die Abschnitte 2.5.1 und 2.5.2), Texture Maps (2.4.6) sowie vielfältiger subtiler Oberflächenstrukturen;
- *Render Room* zur Auswahl der Größe des zu berechnenden Bildes, des Rendering-Verfahrens (einschließlich der Aktivierung globaler Beleuchtung, vgl. 2.5.6) sowie der Qualität der Bildberechnung (u. a. Anti-Aliasing, siehe 2.5.7);
- *Storyboard Room* für die Anzeige und Bearbeitung von Keyframes (siehe S. 156).

In anderen 3D-Grafikprogrammen sind zwar dieselben Aufgabenbereiche zu bearbeiten, die strikte Unterteilung der Arbeitsoberfläche in Räume besteht dort jedoch nicht; vielmehr werden die Aufgaben in Fenstern bearbeitet, die sich bei Auswahl entsprechender Schaltflächen innerhalb des Hauptfensters oder darüber öffnen.<sup>195</sup>

Die bisher vorgestellte Software Carrara bietet, wie bereits die hier gegebenen kurzen und unvollständigen Beschreibungen der Programmfunktionen und -bereiche zeigen, eine große Vielfalt an Möglichkeiten für die Generierung dreidimensionaler Grafiken und Animationen. Dabei handelt es sich um ein Grafikpaket der Mittelklasse,<sup>196</sup> das vor allem in kleineren Design-Firmen verwendet wird. Für die Nutzung in größeren Projekten wie Spielfilmen und umfangreicheren Architekturvisualisierungen sowie für die Produktion aufwändiger Computerspiele kommen Softwarepakete zum Einsatz, deren Preise je nach Ausbaustufe zwischen eintausend und mehreren zehntausend Euro liegen.<sup>197</sup> Derartige professionelle Programme zeichnen sich gegenüber preisgünstigeren Lösungen vor allem durch folgende Merkmale aus:

- Möglichkeit des Einsatzes in Netzwerken und der gleichzeitigen Arbeit mehrerer Mitarbeiter an denselben Projekten;
- Module für NURBS-Flächen und Subdivision Surfaces, welche eine sehr detaillierte Modellierung organischer Objekte ermöglichen (siehe 2.3.6, 2.3.8 und 2.3.9);
- Animationsmodule für Charakter Animation und Soft Body Dynamics (siehe S. 157); Simulation von Haaren und natürlich wirkenden Kleidungsstücken (Cloth Simulation) einschließlich ihrer Beeinflussung durch Wind;

<sup>195</sup>Durch die strikte Abgrenzung der Aufgabenbereiche gilt Carrara im Vergleich zu anderen 3D-Grafikpaketen als recht einfach zugängliche Software. Die Untergliederung in Räume erleichtert die Orientierung in dieser hinsichtlich der Vielfalt der Objekte, der Bearbeitungsschritte sowie der Mannigfaltigkeit von Optionen sehr komplexen Softwarekategorie. Die häufig notwendigen Wechsel zwischen den verschiedenen Räumen können jedoch die Produktivität erfahrener Anwender beeinträchtigen.

<sup>196</sup>Der Preis für Carrara beträgt in Abhängigkeit von der Version ca. EUR 400,- bis 600,-. Für Lehr- und Lernzwecke sind Schulversionen für ca. EUR 100,- erhältlich. Die ältere Version 1.1 (von 1999) lag in den Jahren 2003 und 2004 einigen Zeitschriften bei und darf unbegrenzt und uneingeschränkt genutzt werden. Ihr fehlen allerdings einige Möglichkeiten späterer Versionen, u. a. die globale Illumination.

<sup>197</sup>Zu den preisgünstigeren professionellen Produkten gehören Maxon Cinema 4D ([www.maxon.net](http://www.maxon.net)) und Newtek Lightwave 3D ([www.newtek.com](http://www.newtek.com)). Der Preis von 3ds max liegt bei ca. EUR 5000,- für die kommerzielle Nutzung und ca. EUR 900,- für Lehrzwecke ([www4.discreet.com/3dsmax](http://www4.discreet.com/3dsmax)). Höher sind die Preise der bei Großproduktionen führenden Systeme Softimage XSI ([www.softimage.com](http://www.softimage.com)) und Alias Wavefront Maya ([www.alias.com](http://www.alias.com)), die von ihren Herstellern auch für kundenspezifische Anforderungen konfiguriert werden.



Abbildung 2.92: Boolesche Differenz als nicht-destruktive Modifikation

- Programmierschnittstellen für Scripting und Automatisierung, Möglichkeit der Einbindung verschiedener Renderer wie Mental Ray und RenderMan (s. S. 165);
- Einstellmöglichkeiten für das Rendering, die optimale Kompromisse zwischen Darstellungsqualität und Renderzeit ermöglichen: Nutzung von Raytracing, Radiosity und Photon Mapping für ausgewählte Objekte, Rendern in Texturen, die dann für weitere Bilder verwendbar sind („Texture Baking“).<sup>198</sup>

Ein mitunter sehr bedeutsamer Unterschied zwischen 3D-Grafiksystemen verschiedener Preisklassen besteht darin, so genannte *nicht-destruktive Modifikationen* vornehmen zu können, nach denen die Ausgangsobjekte noch veränderbar sind und sich das Ergebnis mit ihnen vorgenommener Operationen nach Veränderungen anpasst. Als Beispiel zeigt Abb. 2.92 das Ergebnis einer in 3ds max durchgeführten Booleschen Operation (Differenz). Die Ausgangsobjekte (extrudierter Text, Quader) sind weiterhin veränderbar. Wird zum Jahreswechsel der Text geändert, so aktualisiert sich das Objekt automatisch. Im Gegensatz dazu werden bei der Durchführung Boolescher Operationen in Carrara automatisch Polygonnetze generiert, die Ausgangsobjekte sind nicht mehr vorhanden. Zum Jahreswechsel muss das gesamte Objekt neu konstruiert werden, falls nicht vor der Anwendung Boolescher Operationen eine Kopie der Szene gespeichert wurde.

Viele der oben aufgezählten Möglichkeiten professioneller 3D-Grafiksysteme sind für die in der vorliegenden Arbeit zu untersuchenden Anwendungen im Mathematikunterricht nicht sehr relevant. Ein wichtigeres Kriterium ist hierbei die schnelle Erlernbarkeit des Umgangs mit einer Software. Dieses Kriterium wird gerade durch die hoch professionellen Werkzeuge wie Softimage XSI und Alias Wavefront Maya am wenigsten erfüllt; bereits die versierte Nutzung von Teilfunktionen dieser Softwarepakete erfordert sehr lange Einarbeitungszeiten. Durch eine schnelle Einarbeitung und intuitive Bedienbarkeit zeichnet sich am ehesten das (in den vorangegangenen Abschnitten deshalb auch etwas häufiger herangezogene) Programm Carrara aus. Auch im direkten Vergleich mit der (erheblich teureren) Software 3ds max ermöglicht Carrara oftmals mit geringerem Zeitaufwand annähernd gleichwertige Ergebnisse.<sup>199</sup>

<sup>198</sup>Vergleichstests verschiedener 3D-Softwarepakete, in denen die Implementierung dieser und einiger anderer fortgeschrittener Funktionen sowie Bedienbarkeit und Praxistauglichkeit untersucht werden, finden sich u. a. in [165] und [166] (2001), [47] (2003) sowie [40] (2005).

<sup>199</sup>Hinsichtlich des Funktionsumfangs von 3D-Grafiksoftware führt ein Rückblick auf den Zeitraum von 1995 bis Anfang 2006 zu der Feststellung, dass viele Funktionen, die noch vor einigen Jahren nur in sehr teuren Paketen zu finden waren, nunmehr Einzug in preisgünstigere Lösungen oder sogar in freie Software gehalten haben. Dazu zählen Radiosity, Photon Tracing und Caustics im Bereich des Rendering; B-Splines und Subdivision Surfaces im Modellierungs- sowie inverse Kinematik und einfachere Physiksimulationen im Animationsbereich. 3ds max (siehe u. a. [250]) stand für die vorliegende Arbeit in der Version 5 (aus dem Jahr 2002) zur Verfügung. Im Hinblick auf einige Merkmale ging die 2005 aktuelle Version 4 von Carrara bereits darüber hinaus.

Da mathematische Beschreibungen von Flächen im Produktionsbereich kaum zur Anwendung kommen, ermöglichen die meisten 3D-Grafikprogramme (wie z. B. 3ds max) nicht die Darstellung von Flächen durch die Eingabe von Gleichungen bzw. Parameterdarstellungen. Zu den Ausnahmen gehören in dieser Hinsicht die Programme Carrara und Maxon Cinema 4D, welche durch Parameterdarstellungen gegebene Flächen darstellen können.<sup>200</sup> Auf die Darstellung von Parameterflächen in Carrara wurde bereits in Abschnitt 2.3.5 eingegangen (siehe S. 92). Der von Cinema 4D verfolgte Ansatz ist allgemeiner und dadurch etwas schwieriger zu handhaben. Objekte werden hier nicht durch Parameterdarstellungen *erzeugt*, sondern bereits vorhandene Objekte *deformiert*.

Dazu muss zunächst ein Grundobjekt erzeugt und auf dieses dann eine mathematisch beschriebene Deformation angewendet werden. Im einfachsten Falle handelt es sich bei dem Grundobjekt um ein Rechteck in einer Koordinatenebene (das Rechteck wird in dem Eigenschaftsfenster von Cinema 4D als „Ebene“ bezeichnet, siehe Abb. 2.93, obwohl es sich nur um einen Teil einer Ebene handelt). Durch Anwendung eines Deformationsobjekts (Formel-Objekt) lässt sich daraus eine durch die einzugebende Parameterdarstellung beschriebene Fläche erzeugen.

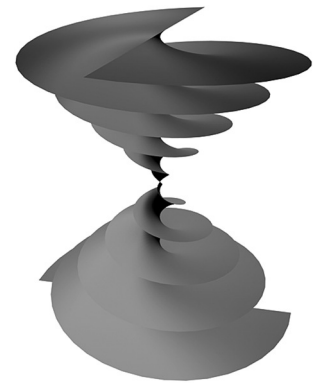
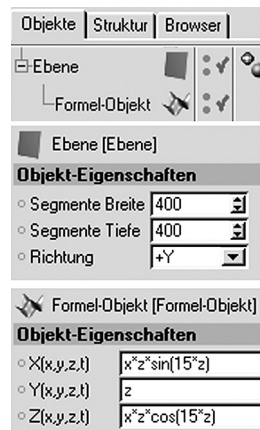


Abbildung 2.93: Parameterdarstellung einer konischen Spirale in Maxon Cinema 4D

Ein Vorteil von Cinema 4D gegenüber Carrara besteht darin, dass die Feinheit der Unterteilung in Intervalle und somit der Annäherung durch Polygonnetze in Cinema 4D frei wählbar und somit eine sehr genaue Darstellung von Flächen möglich ist.<sup>201</sup>

Auch im Bereich der freien Software existieren 3D-Grafikprogramme mit grafischen Benutzeroberflächen; das bekannteste davon ist *Blender*, das über leistungsfähige Modellierungs-, Animations-, und Renderfunktionen verfügt, durch seine von den üblichen Konventionen stark abweichende Bedienung jedoch einen sehr hohen Einarbeitungsaufwand erfordert.<sup>202</sup> Der Umgang mit der Shareware *Moray* ([www.stmuc.com/moray](http://www.stmuc.com/moray)) ist demgegenüber leichter zu erlernen. Moray enthält Module für Modellierung und Texturierung; für das Rendering ist POV-Ray einzubinden, wodurch sich Bilder hoher Qualität generieren lassen. Jedoch erreicht m. E. kein Shareware- oder Freeware-Programm die Leistungsfähigkeit in Verbindung mit intuitiver Bedienbarkeit, welche die ältere Carrara-Version 1.1 aufweist, die bereits kostenlos erhältlich war (vgl. Fußnote 196).

<sup>200</sup>Damit lassen sich auch durch Funktionsgleichungen  $z = f(x, y)$  gegebene Flächen darstellen, indem Koordinaten als Parameter eingesetzt werden, z. B.  $x(u, v) = u$ ,  $y(u, v) = v$ ,  $z(u, v) = f(u, v) = f(x, y)$ .

<sup>201</sup>Der Unterschied wird bei Betrachtung der Abbildungen *Parameterdarstellung-Carrara.jpg* und *Parameterdarstellung-Cinema4D.jpg* in dem Ordner 2-7 (siehe Anhang D, S. 365) deutlich.

<sup>202</sup>Viele Funktionen sind in Blender nur über Tastenkombinationen zu erreichen, die nicht den gängigen Standard-Shortcuts entsprechen. Zu Blender siehe [212] und [www.blender.org](http://www.blender.org).



### 2.7.4 CAD-Systeme

CAD-Systeme werden vor allem für Konstruktionsaufgaben u. a. in der Architektur sowie im Maschinen- und Fahrzeugbau eingesetzt.<sup>203</sup> Obwohl zweidimensionales Konstruieren immer noch die dominierende Arbeitsweise im CAD-Bereich sein dürfte, wurden viele CAD-Programme innerhalb der letzten 10 Jahre um Möglichkeiten der 3D-Darstellung und des dreidimensionalen Konstruierens erweitert. Im Rahmen dieser Arbeit wurde das Programm *Imsi TurboCad* (in der Version 8.2 Standard) untersucht.<sup>204</sup>

In CAD-Programmen stehen umfangreiche *Zeichen- und Konstruktionsmöglichkeiten* zur Verfügung, so kann in TurboCad z. B. ein Kreis auf folgende Weisen erzeugt werden:

- durch Mittelpunkt und Radius,
- durch drei Punkte,
- durch 2 diametrale Punkte,
- durch eine tangierende Gerade und zwei Punkte,
- durch einen tangierenden Kreisbogen und zwei Punkte,
- tangential zu drei Kreisbögen sowie
- durch zwei tangentielle Objekte und einen Punkt.

In CAD-Systemen sind vielfältige *geometrische Konstruktionen* durchführbar. Es können u. a. Strecken und Winkel halbiert oder in beliebig viele kongruente Teilstrecken bzw. -winkel unterteilt sowie Senkrechten und Parallelen konstruiert werden. Im Gegensatz zu Dynamischer Geometriesoftware (DGS) ändern sich jedoch bereits konstruierte Objekte nicht, wenn nachträgliche Veränderungen der erzeugenden Objekte erfolgen.

Grundsätzlich ermöglichen CAD-Programme *maßstäbliches Konstruieren*; dabei wird in TurboCad zwischen Modell- und Papierbereichen unterschieden. In Modellbereichen erfolgen Konstruktionen anhand der Abmessungen der zu konstruierenden Objekte, die sich dann mittels eines wählbaren Maßstabs in Papierbereiche abbilden und ausdrucken lassen. Die Bemaßungen der zu konstruierenden Objekte werden dabei aus den Modellbereichen übernommen. Es stehen unterschiedliche *Bemaßungsfunktionen* zur Verfügung. Um genaues Konstruieren zu ermöglichen, existieren mehrere *Einrast- bzw. Objektfangfunktionen* an Gitterpunkten, Hilfslinien sowie an anderen Objekten.

Moderne CAD-Programme ermöglichen neben ebenen auch *dreidimensionale Konstruktionen*. So können in TurboCad 8.2 Quader, Kugeln, Zylinder, Kegel, polygonale Prismen und Tori sowie mittels Boolescher Operationen Vereinigungen, Durchschnitte und

---

<sup>203</sup>Die Abkürzung CAD steht für Computer Aided Design, wobei „Design“ mehr im Sinne von Entwurf als von Grafikdesign zu verstehen ist.

<sup>204</sup>TurboCad wurde gewählt, da es in der Preisklasse bis EUR 750,- über die umfangreichsten 3D-Funktionen verfügt, vgl. [238]. Darüber hinausgehende Softwarepakete wie Autocad oder Microstation bieten kaum Erweiterungen der geometrischen Möglichkeiten (die bereits in den preisgünstigeren Programmen sehr vielfältig sind), dafür jedoch spezialisierte Funktionen z. B. für Architektur und Maschinenbau sowie Datenbankanbindungen für die Einbindung von Bauteileigenschaften und -preisen verschiedener Hersteller. Es sind zudem Funktionen des CAM (Computer Aided Manufacturing) integriert, die bis hin zur Ansteuerung von Maschinen für die Produktion von Prototypen reichen.

Differenzen dieser Körper erstellt werden. Darüber hinaus ist es möglich, ebene Figuren zu extrudieren und dadurch allgemeine Zylinder (z. B. über Bézierkurven) sowie Rotationskörper zu erzeugen.

Die Konstruktion dreidimensionaler Objekte beginnt in TurboCad innerhalb von Ebenen durch Zeichnen mit der Maus. Abmessungen derjenigen Bestimmungstücke der zu erzeugenden Objekte, die außerhalb der Konstruktionsebene liegen, sind numerisch einzugeben.<sup>205</sup> Die Veränderung der Lage und der Abmessungen einzelner Objekte ist auch nachträglich durch die Eingabe von Größen- bzw. Koordinatenwerten möglich. Dabei stehen mehrere Koordinatensysteme zur Verfügung: Neben dem absoluten Koordinatensystem (Weltkoordinatensystem), das sich auf die gesamte Konstruktion bezieht, besitzt jedes Objekt ein Objektkoordinatensystem (siehe Abschnitt 2.4.2); zusätzlich können benutzerdefinierte Koordinatensysteme erstellt werden.

Dreidimensionale Konstruktionen können in 10 Standardansichten (rechtwinklige Ansichten von oben, unten, links, rechts, vorn und hinten sowie isometrische Ansichten Nordwest, Nordost, Südost und Südwest) sowie in benutzerdefinierten Ansichten (festgelegt durch Projektionsebenen) dargestellt werden (siehe Abb. 2.94). Darüber hinaus verfügt Turbocad über einige Möglichkeiten von 3D-Grafiksystemen wie das Einfügen von Lichtquellen und die Festlegung von Kameras. Für die Bildberechnung stehen Scanline-Rendering-Verfahren und Raytracing zur Verfügung, wobei allerdings nicht die optische Qualität von 3D-Grafiksoftware erreicht wird.

Die Bedienung von CAD-Software zu erlernen, ist aufgrund der mannigfaltigen Optionen derartiger Programme sehr aufwändig. Insbesondere die Nutzung der 3D-Funktionen stellt hohe Anforderungen an Raumvorstellungsvermögen, Software-Bedienung und die Nutzung verschiedener Ansichten, zwischen denen häufig gewechselt werden muss. Um 3D-Szenen zu erstellen und zu rendern, führen einfache 3D-Grafikprogramme wesentlich schneller zum Ziel als CAD-Software. Der Einsatz von CAD-Systemen ist jedoch dann notwendig, wenn exakte Konstruktionsfunktionen wie Objektfang, maßstäbliches Zeichnen und Konstruieren sowie Bemaßungen erforderlich sind.

<sup>205</sup>Eine übliche Vorgehensweise, um in TurboCad z. B. einen Quader zu erstellen, besteht darin, mit der Maus in der Konstruktionsebene ein Rechteck aufzuziehen und die Höhe des Quaders als Zahlenwert einzugeben. Um auf diese Weise verschiedene Objekte in gewünschter Lage zu erzeugen, muss mit mehreren Konstruktionsgrundebenen gearbeitet werden.

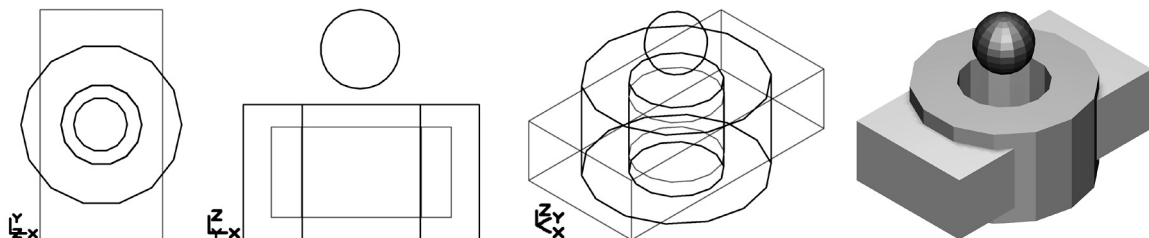


Abbildung 2.94: Grundkörper und Boolesche Operationen in TurboCad

v. l. n. r.: Ansicht von oben (Planmodell), von vorn, isometrische Ansicht (Südost), gerendertes Modell

### 2.7.5 Computeralgebrasysteme

Computeralgebrasysteme (CAS) sind im Bereich der 3D-Computergrafik vor allem für die Darstellung von Flächen interessant.<sup>206</sup> Die wichtigsten mathematischen Beschreibungsformen von Flächen im dreidimensionalen Raum sind

- Funktionsgleichungen der Form  $z = f(x, y)$ ,
- implizite Gleichungen  $f(x, y, z) = 0$  sowie
- Parameterdarstellungen  $x = f_x(u, v)$ ,  $y = f_y(u, v)$ ,  $z = f_z(u, v)$

(siehe Abschnitt 2.3.5). Die Hauptfunktionalität von CAS (nämlich algebraische Umformungen durchzuführen und Gleichungen bzw. Gleichungssysteme zu lösen) wird bei der Darstellung von explizit oder parametrisch gegebenen Flächen nicht genutzt, sondern nur ihre Fähigkeit, Funktionswerte von Stützstellen zu berechnen und die dadurch generierten Gitternetze zu zeichnen.<sup>207</sup> Lediglich bei der Darstellung implizit gegebener Flächen müssen für die Bestimmung von Stützpunkten Gleichungen gelöst werden, was jedoch auch in CAS meist mithilfe numerischer Verfahren geschieht (siehe S. 93).

Für die vorliegende Arbeit wurden die CAS Mathematica und MuPAD etwas ausführlicher sowie darüber hinaus Derive, Maple und Mathcad hinsichtlich ihrer Fähigkeiten untersucht, durch die drei genannten Beschreibungsformen gegebene Flächen darzustellen. Mithilfe von *Derive* ist lediglich die Darstellung durch Funktionsgleichungen und lineare implizite Gleichungen<sup>208</sup> gegebener Flächen, nicht jedoch parametrisch beschriebener Flächen möglich. Die bereits ältere (im Zusammenhang mit den Tafelwerken des Paetec-Verlags recht verbreitete) Version 8 von *Mathcad* erfordert die Erstellung von Matrizen mit Funktionswerten für eine festzulegende Zahl von Stützpunkten, aus denen es Polygonnetze der entsprechenden Flächen generieren kann.<sup>209</sup> In neueren Mathcad-Versionen lassen sich 3D-Grafikdarstellungen aus expliziten Funktionsgleichungen und Parameterdarstellungen unmittelbar generieren.

Die hinsichtlich ihrer grafischen Möglichkeiten leistungsfähigsten CAS sind *Mathematica*, *Maple* und *MuPAD*.<sup>210</sup> Diese Systeme können Flächen auf der Grundlage aller

<sup>206</sup>Außerdem können die Fähigkeiten von CAS hinsichtlich der Berechnung von Lagebeziehungen und Eigenschaften räumlicher Objekte für Aufgaben der Positionierung und Animation genutzt werden.

<sup>207</sup>Hierfür lassen sich auch leistungsfähige Funktionenplotter wie *Turboplot* ([www.turboplot.de](http://www.turboplot.de)) und *GNUPlot* ([www.gnuplot.info](http://www.gnuplot.info)) verwenden. Die Darstellung durch implizite Gleichungen gegebener Flächen beherrschen diese Programme jedoch nicht. Hingegen verarbeitet das Plotprogramm *Autograph* ([www.autograph-math.com](http://www.autograph-math.com)) implizite Gleichungen und stellt auch durch Parameterdarstellungen gegebene Raumkurven, jedoch keine parametrisch gegebenen Flächen dar.

<sup>208</sup>Bereits eine Grafik des durch die Gleichung  $x^2 + 2y^2 + z^2 = 5$  gegebenen Ellipsoids lässt sich in *Derive* nur über den Umweg der Umformung in zwei Funktionsgleichungen  $z(x, y) = \pm\sqrt{5 - x^2 - 2y^2}$  und die Darstellung der dadurch gegebenen beiden Halbellipsoide anfertigen.

<sup>209</sup>Eine MathCad-Datei (*Mathcad8-Flaechendarstellung.mcd*), in der Flächen auf diese Weise dargestellt werden, befindet sich in dem Ordner 2-7. Die Flächen sind interaktiv dreh- und skalierbar. Eine Beschreibung der Flächendarstellung in Mathcad 8 enthält [207], S. 45–52.

<sup>210</sup>Siehe u. a. [173] und [www.mupad.de](http://www.mupad.de) zu MuPad, [www.maplesoft.com](http://www.maplesoft.com) zu Maple, [www.wolfram.com](http://www.wolfram.com) und [140] zu Mathematica sowie [83] speziell zu Animationen in Mathematica.

drei o. a. Beschreibungsformen direkt darstellen<sup>211</sup> sowie animierte Grafiken erzeugen. Hinsichtlich der interaktiven Betrachtung von Grafiken aus verschiedenen Perspektiven haben MuPAD und Maple Vorteile gegenüber Mathematica.<sup>212</sup> Durch Zusatzmodule wie *MathGL3D*<sup>213</sup> oder durch Einbindung des interaktiven Visualisierungswerkzeugs *JavaView* ist jedoch auch für Mathematica eine leistungsfähige Echtzeit-Darstellung gegeben. JavaView lässt sich auch mit Maple und MuPAD nutzen; es ermöglicht zudem die interaktive Betrachtung dreidimensionaler Objekte auf Internetseiten, wobei Parameter verändert werden können und eine Vielzahl weiterer Optionen besteht.<sup>214</sup>

MuPAD benötigt für die Berechnung von Grafiken etwas mehr Zeit als die anderen Systeme, was jedoch auf neueren Computern kaum auffällt und die flüssige Navigation bei der interaktiven Betrachtung nicht beeinträchtigt.<sup>215</sup> Eine Besonderheit von MuPAD ist die Darstellung von Grundkörpern durch Koordinatenbeschreibungen, wodurch dieses System auch als einfache 3D-Szenenbeschreibungssprache nutzbar ist (siehe S. 163).

Eine Besonderheit von Maple (ab Version 9) besteht in der Möglichkeit, Flächen teilweise transparent darzustellen (siehe Abb. 2.95). Auch wenn dabei nicht die Darstellungsqualität von 3D-Grafiksoftware erreicht wird, erweist sich die in Echtzeit erfolgende Darstellung transparenter Flächen vor allem bei Schnittbetrachtungen als nützlich. Mithilfe der oben erwähnten Software JavaView lassen sich auch Objekte, die in MuPAD oder Mathematica generiert wurden, transparent visualisieren (Abb. 2.96).<sup>216</sup>

<sup>211</sup>Beispiele für die Darstellung durch implizite Gleichungen und Parameterdarstellungen gegebener Flächen unter Nutzung von MuPAD und Mathematica enthalten Abschnitt 2.3.5 sowie der Ordner 2-3 (siehe S. 365). Allerdings traten bei der Darstellung einiger implizit gegebener Flächen in Mathematica (3.01 und 4.12) mithilfe der dafür vorgesehenen Funktion `ContourPlot3D` Probleme auf.

<sup>212</sup>Mathematica zeichnet sich andererseits aufgrund der für die Grafikdarstellung verwendeten Postscript-Technologie durch eine besonders gute Druckqualität aus.

<sup>213</sup>Siehe [146] und <http://phong.informatik.uni-leipzig.de/~kuska/mathgl3dv3>.

<sup>214</sup>In den genannten CAS oder in 3D-Grafiksoftware erstellte dreidimensionale Objekte können mithilfe von JavaView nicht nur betrachtet, sondern auch modifiziert werden. So lassen sich Polygonnetze u. a. triangulieren, mithilfe von Subdivision-Algorithmen (siehe S. 111) verfeinern sowie durch das Entfernen von Eckpunkten vergrößern; zu JavaView siehe [30], Kap. 15 und [www.javaview.de](http://www.javaview.de).

<sup>215</sup>Wie bereits in Abschnitt 2.7.2 ausgeführt wurde, nutzt MuPAD für die Darstellung von 3D-Grafiken die OpenGL-Bibliotheken. Vorher muss das CAS jedoch selbst Polygonnetze berechnen, welche die entsprechenden Flächen repräsentieren, und diese an die Open-GL-Schnittstelle übergeben.

<sup>216</sup>Das Grafiksystem von MuPAD ermöglicht ebenfalls die Transparenz einer Reihe von Objekten; allerdings arbeitete die Transparenz zumindest in den Versionen bis 4.02 (2007) nicht zuverlässig, so dass z. B. die transparente Darstellung von parametrisch beschriebenen Flächen nicht möglich war.

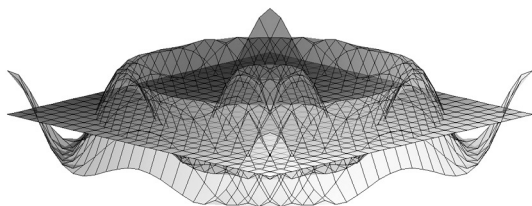


Abbildung 2.95: Transparente Flächen-darstellung in Maple

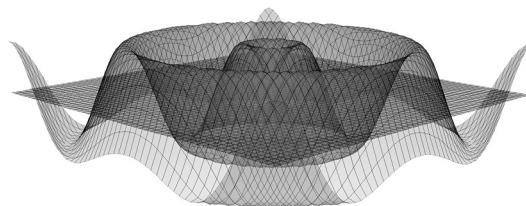


Abbildung 2.96: Transparente Darstellung einer MuPAD-Grafik in JavaView

Die CAS Maple, Mathematica und MuPAD erlauben die Weiterverarbeitung in ihnen erstellter Flächen in 3D-Grafikprogrammen.<sup>217</sup> Der Export von 3D-Grafiken aus CAS ist für die Nutzung in Grafikprogrammen sinnvoll, die nicht über die Funktionalität verfügen, durch Gleichungen bzw. Parameterdarstellungen beschriebene Flächen darzustellen. Dazu gehören 3ds max, ältere Versionen von POV-Ray und viele andere Softwarepakete. Da Cinema 4D, Carrara und POV-Ray (seit der 2002 erschienenen Version 3.5) selbst Flächen, die durch mathematische Beschreibungen gegeben sind, generieren können, sind die Exportmöglichkeiten von CAS hierfür von geringerer Bedeutung.

Außer den Möglichkeiten der grafischen Darstellung durch Gleichungen gegebener Flächen sind für Anwendungen in der Computergrafik vor allem die in CAS wie MuPAD, Mathematica und Maple vorhandenen Programmiersprachen interessant, welche u. a. die Darstellung rekursiv definierter Kurven und Flächen ermöglichen. Beispiele für die Erzeugung rekursiv definierter Splinekurven und Bézierflächen mithilfe von MuPAD enthalten die Abschnitte 2.3.6 und 2.3.8.

### 2.7.6 Software für den Unterricht in analytischer Geometrie

Mit CAS lassen sich die meisten Aufgaben der analytischen Geometrie lösen und auch Visualisierungen dafür anfertigen.<sup>218</sup> Da jedoch gerade die Darstellung sehr einfacher Objekte (wie z. B. Strecken mit gegebenen Endpunkten oder Vektorpfeile) in CAS teilweise etwas aufwändig ist, wurden einige vergleichsweise einfache Programme speziell für die Veranschaulichung von Vektoren (in Pfeildarstellung), Strecken, Geraden, Ebenen sowie teilweise Kugeln und ihrer Lagebeziehungen entwickelt.

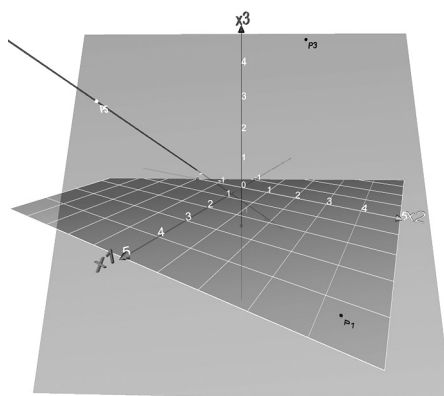
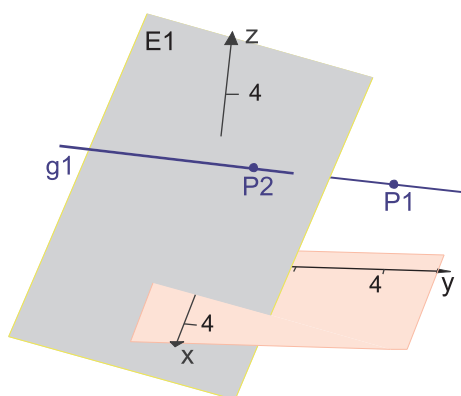
Eine Visualisierungssoftware für Inhalte des Schulstoffes in analytischer Geometrie ist *DreiDGeo* (siehe Abb. 2.97). Diese Software kann Punkte, Strecken, Geraden, Ebenen, Dreiecke und Pyramiden in Echtzeit darstellen, wobei die erzeugten Grafiken interaktiv rotier- und skalierbar sind. Außerdem lassen sich in *DreiDGeo* Schnittpunkte und -geraden, Lote und mittelsenkrechte Ebenen konstruieren, geometrische Abbildungen (Spiegelungen, Streckungen, Projektionen) ausführen sowie Winkel, Längen, Flächeninhalte und Volumina messen.<sup>219</sup> Die Software *Vectory* (siehe Abb. 2.98) beschränkt sich auf Veranschaulichungen von Punkten, Strecken, Geraden, Ebenen und Kugeln, erreicht aber eine gute Qualität der Grafikausgabe und verfügt über gut bedienbare

---

<sup>217</sup>In Mathematica und Maple lassen sich Flächen (als Polygonnetze) in das CAD-Austauschformat DXF exportieren, Maple ermöglicht außerdem einen direkten POV-Ray-Export, der für Mathematica nachrüstbar ist (siehe S. 109). MuPAD kann 3D-Grafiken für die bereits erwähnte Software JavaView exportieren. In JavaView lassen sich 3D-Objekte u. a. in dem verbreiteten Wavefront-Object-Format speichern, das viele 3D-Programme lesen können und dessen Konvertierung in das POV-Ray-Format u. a. mithilfe des 3D-Formatkonverters *PoseRay* (<http://mysite.verizon.net/sfg0000>) möglich ist.

<sup>218</sup>Speziell mit Fragen der (allerdings nur ebenen) analytischen Geometrie in Mathematica befasst sich [255]; zu ebener und räumlicher analytischer Geometrie in MuPAD siehe [89].

<sup>219</sup>Nähere Informationen zu *DreiDGeo* enthalten [3] und [230]; diese Software wurde allerdings seit 2001 nicht mehr weiterentwickelt.

Abbildung 2.97:  
DreiDGeoAbbildung 2.98:  
Vectory

Navigationselemente.<sup>220</sup> Erwähnt sei noch das Programm *Descartes 3D*, welches außer den genannten Objekten auch Kegel und Zylinder darstellt und für einfache Aufgaben der Vektorrechnung genutzt werden kann. Es verfügt ebenfalls über eine recht übersichtliche Grafikdarstellung, wobei die OpenGL-Bibliotheken zum Einsatz kommen.<sup>221</sup>

Die Möglichkeiten der grafischen Darstellung von Objekten in neueren Versionen einiger CAS lassen den Einsatz spezieller Unterrichtssoftware für Visualisierungen in der analytischen Geometrie fragwürdig erscheinen, falls die Schüler ohnehin mit einem dafür geeigneten CAS arbeiten. Als weitere – zunehmend interessanter werdende – Alternative bietet sich seit 2006/07 dynamische Raumgeometriesoftware an.

### 2.7.7 Dynamische Geometriesoftware (DGS) und Software für den Unterricht in Raumgeometrie

Dynamische Geometrie-Systeme (DGS) arbeiten hinsichtlich der Repräsentation, Verarbeitung und Darstellung von Objekten nach dem Prinzip der Vektorgrafik (siehe S. 70). Die Besonderheit dieser Software-Kategorie gegenüber Vektorgrafik- und CAD-Programmen besteht in der dynamischen Abhängigkeit konstruierter Objekte von ihren „Elternobjekten“. Wird z. B. die Mittelsenkrechte einer Strecke konstruiert, so verändert sich diese in DGS, wenn die Endpunkte der Ausgangsstrecke verschoben werden. Weitere wichtige Möglichkeiten von DGS bestehen darin, Konstruktionen „aufzuzeichnen“ und auf andere Objekte anzuwenden<sup>222</sup> sowie „Spuren“ (Kurven, die abhängige Punkte bei Veränderung der „Elternpunkte“ beschreiben) darzustellen.<sup>223</sup> Insbesondere aufgrund der dynamischen Veränderung geometrischer Konstruktionen sind DGS

<sup>220</sup>Vectory wurde als Facharbeitsprojekt eines Schülers des Deutschhaus-Gymnasiums Würzburg erstellt und ist unter [www.didaktik.mathematik.uni-wuerzburg.de/material/programs1.html](http://www.didaktik.mathematik.uni-wuerzburg.de/material/programs1.html) verfügbar.

<sup>221</sup>Informationen zu Descartes 3D und eine Testversion finden sich unter [www.descartes-media.de](http://www.descartes-media.de). Im Gegensatz zu DreiDGeo und Vectory ist Descartes 3D nicht kostenlos.

<sup>222</sup>Diese „Makrofunktion“ (in dem Programm *The Geometer's Sketchpad* als „Custom Tools“ bezeichnet) steht jedoch nicht in allen DGS zur Verfügung und ist auch keine Spezifik von DGS. Die meisten Vektorgrafik- und CAD-Systeme verfügen ebenfalls über derartige Möglichkeiten.

<sup>223</sup>Ein Beispiel für die dynamische Konstruktion einer Spurkurve in Geometer's Sketchpad wurde in Abschnitt 2.3.6 anhand eines Bézierkurvenstücks gegeben.

zu einer der im Mathematikunterricht der Sekundarstufe I am häufigsten verwendeten Softwarekategorien geworden.<sup>224</sup>

Viele DGS, wie z. B. die recht ausgereifte Software *The Geometer's Sketchpad*, ermöglichen es, Punkte durch ihre Koordinaten einzugeben und Längen-, Winkel- und Flächenmaße sowie Koordinaten von Schnittpunkten auszulesen und damit Berechnungen durchzuführen. Dadurch können mit diesen Programmen Visualisierungen und Berechnungen in der zweidimensionalen analytischen Geometrie durchgeführt werden. Darstellungs-, Beschreibungs- und Konstruktionsmöglichkeiten im dreidimensionalen Raum bieten die 2005 aktuellen DGS (mit Ausnahme von Cabri 3D) nicht. Die Nutzung von zweidimensionalen DGS für die Darstellung räumlicher Körper ist jedoch mit Methoden der darstellenden Geometrie möglich; hierfür wurden – mit dem Ziel des Einsatzes im Mathematikunterricht der SI – einige Vorschläge unterbreitet.<sup>225</sup>

Ende 2004 erschien mit *Cabri 3D* das erste DGS für die Raumgeometrie.<sup>226</sup> Mit diesem Programm können sowohl ebene Figuren (Strecken, Geraden, Kreise und Kegelschnitte) als auch räumliche Körper (Polyeder, Zylinder, Kugeln, Kegel) konstruiert werden. Da die Konstruktionen dynamisch erfolgen, verändern sich Objekte, die in Abhängigkeit von anderen Objekten konstruiert wurden, bei Veränderung der „Elternobjekte“. Auch Konstruktionen von parallelen Geraden und Ebenen, Mittelsenkrechten von Strecken (bei denen es sich um Ebenen handelt) sowie von Loten auf Geraden und Ebenen sind möglich. Cabri 3D führt zudem räumliche Kongruenzabbildungen aus.

Eine interessante Möglichkeit in Cabri 3D ist die Konstruktion sich dynamisch verändernder Schnittkurven zwischen Ebenen und Sphären sowie Kegeln (siehe Abb. 2.99). Für die Anzeige nutzt Cabri 3D (wie auch das im Folgenden beschriebene Archimedes Geo3D) die OpenGL-Bibliotheken (siehe S. 161), wodurch eine flüssige interaktive

---

<sup>224</sup>Stellvertretend für die inzwischen sehr umfangreiche Literatur zur Verwendung von DGS im MU sei hier lediglich auf [16], [265] (Kap. V) sowie die Proceedings zu den Tagungen des Arbeitskreises Mathematikunterricht und Informatik der Gesellschaft für Didaktik der Mathematik verwiesen.

<sup>225</sup>ELSCHENBROICH zeigte in [55] Möglichkeiten der Nutzung von DGS für die Konstruktion perspektivischer Bilder von Körpern auf. Konstruktionen zur Darstellung räumlicher Objekte mit *Cabri Géomètre* entwickelte VÁSÁRHELYI (<http://xml.inf.elte.hu/~mathdid/tartalom.html>). Beispiele für die Verwendung des DGS *Zirkel und Lineal* (Z.u.L.) für räumliche Darstellungen demonstriert dessen Autor GROTHMANN (<http://mathsrv.ku-eichstaett.de/MGF/homes/grothmann/java/zirkel/>), merkt jedoch an, dass Z.u.L. als zweidimensionales Programm dafür an sich nicht konzipiert ist.

<sup>226</sup>Bemühungen, ein dreidimensionales DGS zu entwickeln, wurden schon seit längerem von KLEMMENZ unternommen. Allerdings hat das von ihm bereits 2001 angekündigte *Geometer Pro* (siehe [133] und <http://geosoft.ch>) nie das Teststadium verlassen und wurde von 2002 bis 2007 nicht weiterentwickelt. Beispiele für ältere Raumgeometrie-Programme sind *Schnitte* (1994, siehe [225]), das u. a. darauf basierende *Körper* sowie dessen Weiterentwicklung *Körpergeometrie* (1999, siehe [228], [229]). Mithilfe des zuletzt genannten Programms können vielfältige geometrische Körper konstruiert, in verschiedenen Ansichten dargestellt und zu komplexeren Körpern zusammengesetzt werden. Außerdem ist es möglich, Netzdarstellungen von Körpern zu generieren, Messungen durchzuführen sowie Körper zu drehen und zu spiegeln. Es lassen sich Koordinaten von Eckpunkten anzeigen und Körper durch die Eingabe von Koordinaten und Größen festlegen. Weiter gehende analytische Vorgehensweisen sind bei diesem für den Unterricht in synthetischer Raumgeometrie entwickelten Programm jedoch nicht möglich.

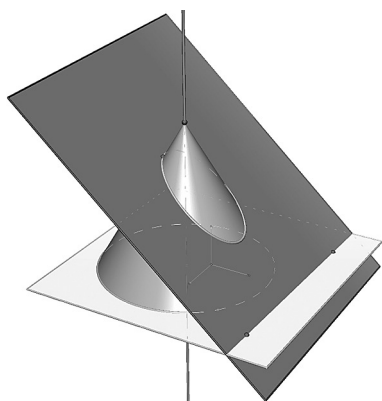


Abbildung 2.99: Kegelschnitt in der Software Cabri 3D

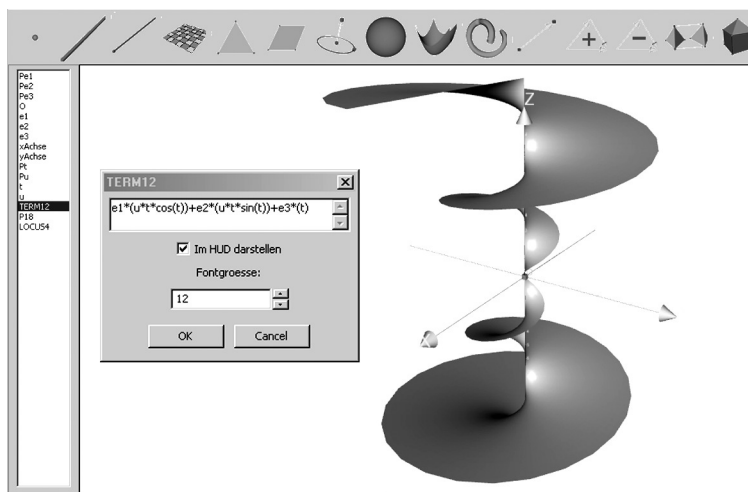


Abbildung 2.100: Konische Spiralfäche in Archimedes Geo3D

Darstellung der Konstruktionen auf dem Bildschirm erfolgt.

Die als Hilfsmittel für die synthetische Raumgeometrie konzipierte Software Cabri 3D ermöglicht (seit der Version 2) den Zugriff auf Koordinaten von Punkten und die Anzeige (nicht jedoch Veränderung) von Gleichungen für zuvor synthetisch erzeugte Geraden, Ebenen und Kugeln. Für Visualisierungen in der analytischen Geometrie ist Cabri 3D daher auf dem gegenwärtigen Stand (2007) nur sehr eingeschränkt nutzbar.

Als erste dynamische Raumgeometriesoftware, mithilfe derer in größerem Umfang auch Veranschaulichungen in der analytischen Geometrie möglich sind, erschien 2006 *Archimedes Geo3D*.<sup>227</sup> Mithilfe dieser Software können u. a. Vektoren durch Anfangs- und Endpunkte festgelegt sowie als Summen oder Differenzen anderer Vektoren konstruiert werden. Des Weiteren ist die analytische Beschreibung von Geraden und Ebenen (durch Parameterdarstellungen und Gleichungen) sowie Kugeln möglich. Darstellbar sind in Archimedes Geo3D auch Flächen als Funktionsgraphen zweier Variabler sowie durch Parameterdarstellungen gegebene Kurven und Flächen (siehe Abb. 2.100).<sup>228</sup> Zudem lassen sich Punktmengen mit konstanten Abstandseigenschaften (Ortslinien und -flächen) in dieser Software intuitiv konstruieren und gut veranschaulichen.

Zusammengefasst kann erwartet werden, dass die gegenwärtige Entwicklung von dynamischer Raumgeometriesoftware, insbesondere des noch sehr jungen Programms Archimedes Geo3D, dazu führt, dass derartige Software auch für Visualisierungen von Gegenständen der analytischen Geometrie in breiterem Umfang Anwendung findet.

<sup>227</sup>Nähere Informationen zu Archimedes Geo3D enthält die Internetseite [www.raumgeometrie.de](http://www.raumgeometrie.de). Dort lässt sich auch eine (zeitlich limitierte) Testversion dieser Software herunterladen.

<sup>228</sup>Diese analytisch beschriebenen Objekte werden durch Makros erzeugt und dargestellt. Die spätere Variation von Gleichungen und Parameterdarstellungen ist (zumindest in den Versionen bis Mai 2007) etwas unübersichtlich, allerdings erfolgt eine schnelle Weiterentwicklung von Archimedes Geo3D.



## Kapitel 3

# Grundsätzliche Überlegungen zur Einbeziehung von Elementen der 3D-Computergrafik in das Stoffgebiet Analytische Geometrie

Bereits in Abschnitt 1.6.2 wurden Potenzen der Einbeziehung von Elementen der Computergrafik in den Unterricht der analytischen Geometrie aufgeführt. Konzepte für die Nutzung dieser Potenzen zu entwickeln, ist Ziel dieses und des folgenden Kapitels. Das vorliegende Kapitel stellt dazu, vorerst in recht allgemeiner Form, Bezüge zwischen den in Kapitel 1 angestellten didaktischen Überlegungen und der in Kapitel 2 vorgenommenen Sachanalyse mathematischer Grundlagen der 3D-Computergrafik her.

Nach einer Zusammenfassung in den vergangenen 20 Jahren unterbreiteter Vorschläge zur Behandlung von Elementen der Computergrafik im Unterricht der analytischen Geometrie in Abschnitt 3.1 wird auf mit der Einbeziehung von Elementen der Computergrafik verbundene Ziele und mögliche Herangehensweisen eingegangen (3.2).

Die Aufnahme von Elementen der Computergrafik führt zu ergänzenden Sichtweisen auf einige zentrale Gegenstände des Unterrichts in analytischer Geometrie. Sich daraus ergebende didaktische Konsequenzen und Möglichkeiten hinsichtlich der Arbeit mit Vektoren und mit Parameterdarstellungen werden in Abschnitt 3.3 erörtert.

Eine Voraussetzung für die Entwicklung konkreter Vorschläge hinsichtlich der Einbeziehung von Elementen der 3D-Computergrafik in den Unterricht, die Gegenstand von Kapitel 4 sind, bildet die Auswahl einer geeigneten Software. Dazu werden am Ende des vorliegenden Kapitels prinzipiell in Frage kommende Softwarekategorien bezüglich ihrer Eignung für die Verwendung im Stoffgebiet Analytische Geometrie verglichen (3.4), Voraussetzungen für die Nutzung der dabei favorisierten Software diskutiert sowie geeignete Anleitungen und Vorlagen, die Schülern die Nutzung der Software erleichtern können, beschrieben (3.5).

### 3.1 Überblick über Vorschläge zur Einbeziehung von Elementen der Computergrafik in den Mathematikunterricht

Beziehungen zwischen der analytischen Geometrie und der Computergrafik sind seit längerem Gegenstand didaktischer Überlegungen und entsprechender Vorschläge. In einer umfassenden Arbeit untersuchte GIEDING bereits 1988 Möglichkeiten der Einbeziehung von Elementen der Computergrafik in den Mathematikunterricht (siehe [90]). Er entwickelte Konzepte und Aufgabenvorschläge zu den Themengebieten

- Rasterung analytisch beschriebener Objekte (Strecken und Kreise) für die Bildschirmdarstellung (vor allem BRESENHAM-Algorithmus);<sup>1</sup>
- Beschreibung geometrischer Abbildungen (Verschiebungen, Drehungen und Skalierungen) durch Matrizen; Zentral- und Parallelprojektionen;
- Algorithmen zur Bestimmung der Sichtbarkeit von Kanten und Flächen;
- Beschreibung ebener Figuren (vor allem Polygone) durch Koordinaten und ihre Darstellung auf dem Bildschirm;
- Generierung ästhetisch ansprechender Grafiken durch Algorithmen;
- Algorithmen zur Darstellung ebener sowie räumlicher Kurven und Flächenstücke.

Das Paradigma, unter dem er die Einbeziehung der Computergrafik in den Unterricht der analytischen Geometrie betrachtete, beschrieb GIEDING wie folgt ([90], S. 101):

*„Der Hauptinhalt des Arbeitens mit Elementen der Computergrafik wird im Anwenden geometrischen Wissens und Könnens durch die Schüler gesehen, wobei dieses Anwenden weitestgehend unter algorithmischen Aspekten erfolgt.“*

Als wichtige Gesichtspunkte seiner Vorschläge nannte er u. a.

- „Zugänge zu neuen geometrischen Objekten“,
- „Konkretisierung und Erlebbarkeit der Geometrie“,
- „Schulung im mathematischen Modellieren“,
- „Erlebbarkeit von Schönheit und Ästhetik der Mathematik“ sowie
- „Möglichkeiten entdeckenden Lernens“ ([90], S. 148f.);

mithin also sämtlich Aspekte, die auch heute bei der Computernutzung im Unterricht der analytischen Geometrie im Mittelpunkt stehen (vgl. 1.5). Die Realisierung der Vorschläge von GIEDING erforderte zur damaligen Zeit die Programmierung der verwendeten Prozeduren in einer Programmiersprache.<sup>2</sup> Dies führt allerdings zu einem recht hohen zeitlichen Aufwand<sup>3</sup> und setzt für eine realistische Umsetzung im Stoffgebiet

<sup>1</sup>Einen Unterrichtsvorschlag zum BRESENHAM-Algorithmus unterbreitete auch SCHWILL, vgl. [236].

<sup>2</sup>Hauptsächlich wurden an Schulen seinerzeit Pascal und Basic verwendet. GIEDING entwickelte in seiner Arbeit Pascal-Prozeduren für alle von ihm vorgeschlagenen Inhalte.

<sup>3</sup>Um den Zeitaufwand zu reduzieren, schlug GIEDING vor, dass Schüler unvollständige Prozeduren, die ihnen vorgegeben werden, um aus mathematischer Sicht interessante Bestandteile ergänzen.

Analytische Geometrie voraus, dass die Schüler bereits mit einer Programmiersprache vertraut sind, was jedoch i. Allg. nicht gegeben war und es auch heute nicht ist.

Eine von GIEDING an einigen Stellen seiner Arbeit vorgesehene Alternative zur Nutzung einer Programmiersprache besteht darin, die Wirkungsweise von Verfahren der Computergrafik (wie Sichtbarkeitsbestimmungen) für die Motivierung von Inhalten der analytischen Geometrie (wie Schnittpunkt- und Normalenberechnungen) zu nutzen, wobei die Schüler entsprechende Überlegungen und Berechnungen dann aber ohne Verwendung des Computers anstellen müssten. Wie bereits in Kapitel 1 erwähnt wurde, ist die motivierende Funktion von Anwendungen mathematischer Inhalte jedoch gering, wenn diese lediglich genannt werden. Um Inhalte der analytischen Geometrie mit deren Anwendungen in der Computergrafik überzeugend zu motivieren, ist es notwendig, dass die Schüler mit computergrafischen Anwendungen arbeiten und die Nützlichkeit des Erkennens ihrer „mathematischen Funktionsweise“ bei dieser Arbeit erleben können. Die Voraussetzungen dafür waren nicht in ausreichendem Maße gegeben, so dass die Vorschläge von GIEDING nur sehr punktuell wirksam werden konnten.

Ende der achtziger und Anfang der neunziger Jahre erschien eine Reihe von Beiträgen, zur Einbeziehung der Computergrafik in den Unterricht, die in ihren grundlegenden Ansätzen viele Gemeinsamkeiten mit den Vorschlägen von GIEDING aufweisen.

- G. BIELIG-SCHULZ und CH. SCHULZ veröffentlichten 1988 zwei Beiträge, deren Kerninhalte Sichtbarkeitsbestimmungen und Projektionen darstellen.<sup>4</sup> In [24] behandelten sie diese Themen vorrangig aus der Sicht der analytischen Geometrie und in [25] gingen sie hauptsächlich auf algorithmische Aspekte sowie geeignete Datenstrukturen für die 3D-Grafik ein. Für die Behandlung der Thematik sahen sie vor allem fächerübergreifende Projekte als geeignet an.
- K. ENDL verfasste in der zweiten Hälfte der achtziger Jahre einige in die 2- und 3-dimensionale Computergrafik einführende Bücher,<sup>5</sup> zu deren Hauptbestandteilen Grundlagen aus der analytischen Geometrie und Erläuterungen bei der programmiertechnischen Umsetzung (in der Sprache Pascal) verwendeter Grafikprozeduren gehören. In [61] unterbreitete ENDL Vorschläge zur Darstellung ebener Kurven im Mathematikunterricht, wobei Fragen der Programmierung ebenfalls breiten Raum einnahmen.
- Für den Informatikunterricht erschien 1995 das Arbeitsheft [108] von HEISS und HERMES, das die in Pascal zu erstellenden grundlegenden Prozeduren (vor allem Projektionen und Sichtbarkeitsuntersuchungen) zur Darstellung dreidimensionaler Objekte in recht kompakter Weise darstellt sowie u. a. die Umwandlung von Flächen in Polygonnetze und die Erzeugung fraktaler Landschaften beinhaltet.

---

<sup>4</sup>Bereits 1987 erschien das Buch „3D-Graphik in Pascal“ ([23]) dieser beiden Autoren, das die fachliche Grundlage der in den genannten Beiträgen angestellten didaktischen Überlegungen bildete.

<sup>5</sup>Siehe [59], [60] und [62]; unmittelbare Bezüge zum Unterricht sind in diesen Büchern nicht enthalten, ENDL empfahl jedoch ihre Nutzung durch Schüler und Studenten.

- WUNDERLING schlug 1989 in [280] die Behandlung homogener Koordinaten und des Matrizenkalküls mit  $4 \times 4$ -Matrizen als wichtige mathematische Grundlagen der 3D-Computergrafik vor. Abbildungsgeometrie mit Matrizen als Grundlage der Computergrafik ist auch Gegenstand einer Reihe weiterer Vorschläge, u. a. von LEHMANN (siehe z. B. [149], [152]) sowie von WELLER ([266]), wobei in diesen Beiträgen (1993–1997) nicht mehr die Verwendung einer Programmiersprache, sondern eines Funktionenplotters bzw. des CAS Derive vorgeschlagen wurde.
- Ein Arbeitsbuch zur Computergrafik und darstellenden Geometrie im Unterricht der linearen Algebra ([56]) von ELSCHENBROICH und MEINERS erschien 1994. Es beinhaltet die bereits mehrfach erwähnten Projektionen und Sichtbarkeitsbestimmungen, wobei der Schwerpunkt auf der Anwendung von Matrizen liegt. Die Implementierung der im Buch beschriebenen Verfahren in einer Programmiersprache ist nicht Gegenstand des Buches; stattdessen liegt eine Software bei, unter deren Nutzung diese Verfahren angewendet werden können.

Zusammenfassend kann für die meisten „frühen“ Vorschläge (bis ca. 1995) zur Behandlung der 3D-Computergrafik festgestellt werden, dass Projektionen und Sichtbarkeitsuntersuchungen als notwendige Schritte der Bildschirmdarstellung dreidimensionaler Objekte zentrale Bestandteile bildeten. Mit Abbildungsmatrizen und homogenen Koordinaten für die effektive Beschreibung von Zentralprojektionen wären nach vielen dieser Vorschläge mathematische Inhalte in den Unterricht aufgenommen worden, die ansonsten kaum behandelt wurden. Hinzu kam die Notwendigkeit, eine Programmiersprache zu verwenden, deren Beherrschung zu keiner Zeit bei einer Mehrheit von Schülern vorausgesetzt werden konnte.<sup>6</sup> In diesen aufwändigen Voraussetzungen ist m. E. eine wesentliche Ursache dafür zu sehen, dass die genannten Vorschläge nicht in größerem Umfang die Unterrichtspraxis des Stoffgebietes Analytische Geometrie beeinflussten.

Mitte der neunziger Jahre hatten sich die Rechenleistungen und die computergrafischen Möglichkeiten verbreiteter Personalcomputer so weit entwickelt, dass annähernd fotorealistische Darstellungen möglich wurden. Computerspiele erschienen und wurden schnell populär, welche diese Möglichkeiten nutzten. Durch die Medien wurde der Begriff „3D-Computergrafik“ in einem fotorealistischen und effektreichen (z. T. auch effektüberladenen) Sinne geprägt,<sup>7</sup> der für die Schüler wahrscheinlich nur noch schwierig mit den Drahtgitter- und einfachen Flächendarstellungen in Pascal programmierter Grafikprozeduren in Einklang zu bringen war. Gleichzeitig veränderten sich die Voraussetzungen hinsichtlich der Software, die für die Erstellung dreidimensionaler computergrafischer Darstellungen zur Verfügung stand: CAS sowie (auch dreidimensionale)

<sup>6</sup>Bekanntermaßen wurden nach der Verfügbarkeit von CAS nur noch wenige Vorschläge zur Nutzung von „klassischen“ Programmiersprachen im Mathematikunterricht unterbreitet. Die Einarbeitung in CAS (auch in die Programmierung kleinerer Prozeduren mithilfe von CAS) sowie in hinsichtlich spezieller Anwendungen (z. B. der Computergrafik) spezialisierte Skriptsprachen erfordert erheblich weniger Zeit als das Erlernen einer universellen Programmiersprache wie Pascal.

<sup>7</sup>Erinnert sei an den Film „Jurassic Parc“, der sicherlich einer der ersten großen „Meilensteine“ des Eindringens der 3D-Computergrafik in die Medien war.

Funktionenplotter ermöglichten es fortan, Kurven und Flächen des Raumes auf einfache Weise darzustellen. Die Notwendigkeit, mithilfe von Programmiersprachen Prozeduren zu erstellen, um räumliche Objekte zu visualisieren, bestand nicht mehr – die vorher notwendige Koppelung von Visualisierungen mit der Thematisierung von Elementen der Computergrafik entfiel also.<sup>8</sup> Außerdem wurden für übliche PCs kommerzielle Softwarepakete zur Erstellung von 3D-Computergrafiken und -videos verfügbar, die noch Anfang der neunziger Jahre nur auf teuren Grafikworkstations nutzbar waren. Auch im Bereich preiswerter und freier Software erschienen die ersten fotorealistischen 3D-Grafikprogramme bzw. erreichten einen Reifegrad, der ihren Einsatz populär machte.<sup>9</sup>

Wahrscheinlich ist es sowohl auf die begrenzte Wirksamkeit der vorherigen Vorschläge zur Behandlung der Computergrafik im Mathematikunterricht als auch auf die geschilderten Veränderungen der hard- und softwaremäßigen Voraussetzungen zurückzuführen, dass Mitte der neunziger Jahre ein Paradigmenwechsel erfolgte. Die meisten seither unterbreiteten Vorschläge zur Einbeziehung von Elementen der 3D-Computergrafik<sup>10</sup> in den Unterricht beinhalten nicht mehr die Programmierung von Grafikprozeduren, sondern setzen die Nutzung einer Grafiksoftware voraus. Dieser Wechsel der Herangehensweise wird u. a. anhand folgender Bemerkung von MÜLLER (1995) deutlich:

*„Die Programmierung realistischer Darstellungen ist für schulische Verhältnisse recht aufwendig ...; die Ergebnisse sind, obgleich durch eigene Anstrengung gewonnen und damit motivierend, im Vergleich mit den photorealistischen Darstellungen in den Medien recht dürftig.“* ([185], S. 38)

Probleme bei der Behandlung der in den oben genannten Beiträgen vielfach vorgeschlagenen Grundlagen der Computergrafik konstatierte 1997 die Arbeitsgruppe „Computergrafik als Bindeglied zwischen Mathematik- und Informatikunterricht“ des Arbeitskreises „Mathematikunterricht und Informatik“ der GDM und führte dazu an, dass z. B. das Problem der verdeckten Linien oder Flächen in 3D-Grafiken oder die effiziente Ausgabe eines Kreises auf dem Bildschirm zwar interessante mathematische und informatische Fragestellungen aufwerfen, diese aber *„oft abseits des Lehrplans ... (liegen) oder auch so schwierig (sind), dass es nicht gerechtfertigt erscheint, solche Fragestellungen mit allen Lernenden zu bearbeiten“* ([103], S. 178). Als interessanter für die Herstellung

---

<sup>8</sup>Dass die neuen Möglichkeiten schnell ausgiebig genutzt wurden, um Kurven und Flächen in der analytischen Geometrie zu untersuchen, wurde bereits in Abschnitt 1.4.5 deutlich.

<sup>9</sup>Zu den damals verfügbaren freien fotorealistischen 3D-Grafikprogrammen zählten Polyray, Ray-Shade und POV-Ray. Allerdings verfügte POV-Ray noch nicht über alle Möglichkeiten, die es heute bietet. Die Darstellung durch Gleichungen oder Parameterdarstellungen beschriebener Flächen ist z. B. erst seit 2002 mit dieser Software möglich. Ebenfalls ab Mitte der neunziger Jahre erschienen grafische Benutzerschnittstellen wie Moray für Polyray und POV-Ray, mithilfe derer diese Programme ähnlich bedient werden konnten wie teure kommerzielle 3D-Grafikpakete (siehe Abschnitt 2.7).

<sup>10</sup>In diesem Abschnitt werden nur Ansätze betrachtet, Aspekte der 3D-Computergrafik im Unterricht zu *thematisieren* und keine Vorschläge der reinen *Nutzung* von Grafiksoftware bzw. CAS zu Visualisierungszwecken erörtert (siehe dazu u. a. die Abschnitte 1.4.5 und 1.5). Beide Aspekte der Einbeziehung der 3D-Computergrafik (als Werkzeug und als Unterrichtsgegenstand) in den Unterricht werden jedoch im weiteren Verlauf der vorliegenden Arbeit von Bedeutung sein.

von Bezügen zwischen Mathematik- und Informatikunterricht sah diese Arbeitsgruppe *3D-Grafik, Animationen und virtuelle Realität* an, führte dies jedoch nicht näher aus. GUDERIAN hatte bereits 1990, ausgehend von Überlegungen zum Verhältnis zwischen Mathematik und Kunst, die Beschäftigung mit mathematischen Grundlagen der Computeranimation als Schwerpunkt bezeichnet, der „*das besondere fachwissenschaftlich-fachdidaktische Interesse verdient hätte*“ ([98], S. 67).

MÜLLER schlug in [185] hauptsächlich die Verwendung der Software POV-Ray vor, bezog sich dabei allerdings auf den Informatikunterricht. Den Vorteil der vergleichsweise einfach verständlichen Funktionsweise des Raytracing-Verfahrens<sup>11</sup> bei gleichzeitig hochwertigen Resultaten (die allerdings mit auf Mitte der neunziger Jahre verfügbaren Computern erheblichen Bildberechnungszeiten erkaufte wurden) stellte 1995 HABERSTROH heraus (vgl. [101]). MAJEWSKI plädierte 1997 für den Einsatz von POV-Ray im Mathematik- und Physikunterricht, wobei er u. a. von folgender Überlegung ausging:

„*What kind of programs should we use in teaching geometry or physics, describing the nature of real things, their colors, textures and other properties?*“ ([172], S. 212)

Er beschrieb Einsatzmöglichkeiten von POV-Ray u. a. hinsichtlich der analytischen Beschreibung geometrischer Grundkörper sowie von Flächen 2. und höherer Ordnung,<sup>12</sup> der Untersuchung von Schnitten, Vereinigungen und Differenzen von Körpern sowie algorithmisch erzeugter dreidimensionaler Szenen. Sein Fazit lautete (vgl. [172], S. 222):

„*PovRay creations can help with the achievement of various teaching objectives by:*

1. *Increasing the student's interest and motivation.*
2. *Focusing the student's attention on specific 3D concepts.*
3. *Developing interest in geometry, physics and art.*
4. *Developing scientific thinking, problem solving and a scientific method of investigation.*“

Weitere (recht punktuelle) Vorschläge zur Verwendung von POV-Ray im Mathematikunterricht, die auf Unterrichtserfahrungen basieren und bei denen die geometrische Modellierung dreidimensionaler Szenen im Vordergrund steht, wurden u. a. 2004 von LEHMANN ([153], S. 158) und 2005 von KRUMPE ([143], S. 394ff.) unterbreitet.<sup>13</sup> Hervorhebung fanden mehrfach das Motivierungspotenzial der Modellierung mithilfe einer fotorealistischen Grafiksoftware und die Möglichkeiten der Veranschaulichung analytisch beschriebener Objekte im Raum:

„*POV-Ray has great potential in secondary mathematics classrooms. The possibilities are limited only by the creativity and imagination of the teacher and students. Almost unanimously,*

<sup>11</sup>Wie bereits in Kap. 2 ausgeführt wurde, ist das Raytracing-Verfahren bis heute und auf absehbare Zeit das wichtigste Bildberechnungsverfahren der fotorealistischen 3D-Computergrafik. Die Funktionsweise dieses Verfahrens ist im Vergleich zu der anderer Bildberechnungsverfahren der 3D-Computergrafik durch die vergleichsweise wenigen Schritte recht einfach verständlich.

<sup>12</sup>Die Darstellung durch Polynome beschriebener Flächen war in POV-Ray bereits weit vor der Implementation von Flächenbeschreibungen durch beliebige Gleichungen möglich.

<sup>13</sup>Eine Linksammlung zu Unterrichtsprojekten, die mit POV-Ray durchgeführt wurden, enthält die Internetseite [www.f-lohmueller.de/links/index\\_sd.htm#school](http://www.f-lohmueller.de/links/index_sd.htm#school) von LOHMÜLLER, der auch umfangreiche deutschsprachige Tutorials zur Nutzung dieser Software erstellt hat.

*my students have responded favorably to their work with POV-Ray. Besides helping strengthen students visualization of three-dimensional geometry and use of formal geometric properties of these shapes, using POV-Ray has been a lot of fun.*“ ([143], S. 398)

Mathematisch orientierte Projekte zur 3D-Computergrafik unter Verwendung von POV-Ray, die mit besonders interessierten Schülern durchgeführt wurden, beschrieben u. a. OLDENBURG in [191] und LEUDERS in [159]. Beide Autoren stellten neben der geometrischen Modellierung die Erstellung von *Animationen* in den Vordergrund, welche die parameterabhängige Beschreibung von Koordinaten oder von geometrischen Transformationen erfordert (siehe Abschnitt 2.6). LEUDERS hielt die Realisierung dieses Ansatzes auch im regulären Unterricht für möglich und nannte folgende Potenzen hinsichtlich dessen, „*was Schüler dabei lernen können*“:

1. *Anwendung elementargeometrischer Begriffe: Berechnung von Körpern, Ähnlichkeitsgeometrie, Trigonometrie, Kreis- und Kugelgeometrie;*
2. *Lineare Vektorgeometrie: Normalformen von Ebenen, Parameterdarstellungen von Geraden, Abstände im Raum, Schnittgebilde;*
3. *Operieren mit Mengen als räumlichen Gebilden (Schnitt, Vereinigung, etc.);*
4. *Parametrisierte Raumkurven (z. B. parametrisierte Kreislinie);*
5. *Modellieren realer Gebilde und Bewegungen;*
6. *Arbeitsteilung und Kooperation in einer Projektstruktur;*
7. *Einige grundlegende Methoden des Programmierens (Bedingungen, Schleifen, Makros, objektorientierte Denkweise) – diese eher informatikbezogenen Aspekte können auch gänzlich unberücksichtigt bleiben.*“ ([159], S. 17)

## Anmerkungen

Die koordinatengestützte Modellierung dreidimensionaler Objekte und Szenen sowie die Erstellung von Animationen durch parameterabhängige Beschreibungen sind m. E. zentrale Ansatzpunkte der Einbeziehung von Elementen der 3D-Computergrafik in den Unterricht der analytischen Geometrie. Damit werden auch zwei mathematische Aspekte der „Funktionsweise“ der Computergrafik Gegenstände des Mathematikunterrichts. Es besteht jedoch noch eine „Lücke“ zwischen den Ende der achtziger / Anfang der neunziger Jahren angestellten Überlegungen zur Thematisierung von mathematischen Grundlagen der 3D-Computergrafik (auf dem damaligen Stand) und den in den vergangenen zehn Jahren unterbreiteten Vorschlägen zur Nutzung von fotorealistischer 3D-Grafiksoftware im Unterricht. Das Motivierungs- und Visualisierungspotenzial dieser Vorschläge ist zwar evident; Konzepte, anhand der Arbeit mit dieser Art von Software mathematische Grundlagen der Bildberechnung in der fotorealistischen 3D-Computergrafik aufzugreifen und sie für die Motivierung von Inhalten der analytischen Geometrie zu nutzen, fehlen jedoch bislang. Ein wichtiges Ziel der vorliegenden Arbeit besteht deshalb darin, ausgehend von der Nutzung von CG-Software zumindest punktuell<sup>14</sup>

<sup>14</sup>Eine möglichst vollständige Einbeziehung aller wichtigen Schritte der Bildberechnung in der 3D-Computergrafik in den Mathematikunterricht erscheint nicht realistisch. Auch die Programmierung von 3D-Grafiksoftware ist aus den bereits genannten Gründen nicht ernsthaft in Erwägung ziehen.

mathematische Grundlagen der Bildberechnung zu thematisieren und davon ausgehend Begriffe und Verfahren der analytischen Geometrie zu motivieren und zu behandeln sowie auf die Erstellung von Computergrafiken „rückanzuwenden“.

### **Exkurs: Kurse zur Einführung in die Computergrafik an Hochschulen**

In Einführungskursen in die Computergrafik an Universitäten stehen natürlich völlig andere Ziele im Vordergrund als bei der Einbeziehung von Elementen der Computergrafik in den Mathematikunterricht der Schule: Die Studierenden sollen auf die Erstellung computergrafischer Anwendungen vorbereitet werden; benötigte mathematische Grundlagen sind dabei „Mittel zum Zweck“. Dennoch ist ein Blick auf die Inhalte derartiger Kurse nicht uninteressant.

Traditionelle Einführungskurse in die Computergrafik legten die Schwerpunkte auf grafische Datenstrukturen, Rasterisierungsalgorithmen für Grafikprimitive (wie z. B. Strecken und Kreise), Koordinatentransformationen, Projektionen und Sichtbarkeitsuntersuchungen. Farben und Texturierungsmodelle wurden frühestens am Ende des Semesters diskutiert (siehe [281], S. 186).

Mitte der neunziger Jahre bekundeten Dozenten (u. a. OWEN et al. in [196] und XIANG in [281]) ihre Unzufriedenheit mit dem traditionellen Curriculum, da es nicht mehr dem Stand entsprach, den die Computergrafik zu diesem Zeitpunkt erreicht hatte. Vorgeslagen wurden Top-Down-Ansätze, die den Studierenden schnell zu Einblicken in die fotorealistische Computergrafik verhelfen. Die genannten Autoren führten Computergrafikkurse durch, deren Schwerpunkte u. a. Licht und Farben, Beleuchtungsmodelle, Raytracing, geometrische Transformationen, Texturprojektion und Rasterisierungsalgorithmen sowie z. T. auch parametrisierte Kurven und Flächen (vor allem Bézier- und B-Splinekurven sowie -flächen) bildeten. XIANG konstatierte:

*„The students greeted the new syllabus with great enthusiasm. They spent long hours working on their rendering program and were fascinated by the degree of sophistication and the realistic look of their own images.“* ([281], S. 188)

Zur Jahrtausendwende hatte sich der neue Ansatz zur Behandlung der Computergrafik in Einführungskursen weitgehend etabliert, wie aus dem Überblick [278] von WOLFE hervorgeht. LOWTHER et al. ergänzten den Ansatz um Aspekte der Animation und der Nachbearbeitung von Bildern und Videos (Post Processing), siehe [168].

Auch wenn Vorgehensweisen in Hochschulkursen zur Computergrafik aufgrund anderer Zielrichtungen nicht auf den Schulunterricht übertragen werden können, erscheint es interessant, dass hinsichtlich der Gestaltung dieser Kurse zur selben Zeit ein Paradigmenwechsel erfolgte wie bezüglich der Einbeziehung der Computergrafik in den Mathematik- und Informatikunterricht.



## 3.2 Ziele und Herangehensweisen der Einbeziehung von Elementen der 3D-Computergrafik

In Kapitel 2 wurde deutlich, dass in der 3D-Computergrafik die meisten der im gegenwärtigen Unterricht behandelten Inhalte der analytischen Geometrie unmittelbar angewendet werden. Zudem sind diese Inhalte für ein grundlegendes Verständnis einer Reihe von Arbeitsschritten und Verfahren der Computergrafik ausreichend. Es wurde weiterhin sichtbar, dass mittels analytischer Beschreibungen geometrischer Objekte die Erzeugung interessanter Grafiken möglich ist. Zudem gehören „Produkte“ der 3D-Computergrafik mittlerweile zum Alltag und üben auf viele Jugendliche ein hohes Maß an Faszination aus.<sup>15</sup> Die Computergrafik besitzt also für den Unterricht in analytischer Geometrie Visualisierungs-, Anwendungs- und Motivierungspotenziale, deren Nutzung dazu beitragen kann, den oft stark auf die Abarbeitung formaler Kalküle orientierten Unterricht attraktiver und anschaulicher zu gestalten, in stärkerem Maße inhaltliche Vorstellungen von Objekten und Methoden der analytischen Geometrie bei den Schülern auszubilden sowie eine Reihe der in Abschnitt 1.4 beschriebenen Defizite des gegenwärtigen Unterrichts zumindest abzumildern.

### 3.2.1 Übergreifende Ziele

Ausgehend von den in Kapitel 1 angestellten Überlegungen zu den Aufgaben und Defiziten des Stoffgebietes Analytische Geometrie und den in Kapitel 2 deutlich gewordenen Bezügen zwischen der analytischen Geometrie und der Computergrafik ergeben sich die im Folgenden aufgeführten übergreifenden Ziele, die mit der Einbeziehung von Elementen der 3D-Computergrafik in den Unterricht verbunden werden sollten.<sup>16</sup> Auf verschiedene Inhaltsbereiche des Stoffgebietes bezogene Konkretisierungen dieser Ziele werden im Zusammenhang mit der Entwicklung von Unterrichtsvorschlägen und -konzepten in Kapitel 4 vorgenommen.

#### Entwicklung von Anschauungsvermögen und Raumvorstellung<sup>17</sup>

- Herausbildung anschaulicher Vorstellungen analytisch beschriebener geometrischer Objekte bei den Schülern und einer „koordinatenbezogenen Raumvorstellung“ (siehe Abschnitt 1.4.4) auf der Grundlage aktiven Operierens mit Beschreibungen

---

<sup>15</sup>Diese Tatsache wurde in den in Abschnitt 3.1 erwähnten Beiträgen mehrfach herausgestellt und zeigte sich auch in Unterrichtsversuchen, auf die in Kapitel 5 eingegangen wird, sehr deutlich.

<sup>16</sup>Potenzen und Ziele der Computergrafik für den Unterricht in analytischer Geometrie wurden bereits in Abschnitt 1.6.2 genannt; sie werden hier auf der Grundlage von Kapitel 2 und Abschnitt 3.1 präzisiert. Dabei erfolgt lediglich eine Aufzählung ohne nähere Begründungen der genannten Ziele. Die Legitimation der aufgeführten Ziele aufgrund ihrer Bedeutung für die Weiterentwicklung des Unterrichts in analytischer Geometrie wird, soweit sie nicht unmittelbar einsichtig ist, im Kontext der in Abschnitt 1.4 angestellten Überlegungen deutlich.

<sup>17</sup>Die Kategorien, nach denen die Ziele gegliedert sind, sollen lediglich eine schwerpunktmäßige Einordnung ermöglichen. Einige der aufgeführten Ziele enthalten Komponenten, die hinsichtlich mehrerer Aspekte von Bedeutung sind. So lassen sich z. B. anschauliche Vorstellungen, inhaltliches Verständnis und heuristische Strategien nicht vollständig voneinander trennen.

durch Koordinaten und Gleichungen sowie der Anfertigung von Visualisierungen mithilfe einer geeigneten Software.

### **Herausbildung eines umfassenden Verständnisses von Objekten und Methoden der analytischen Geometrie**

- Hervorrufen reichhaltiger, durch Anwendungen in der Computergrafik und -animation unterstützter, Grundvorstellungen von Gegenständen der analytischen Geometrie (vgl. 1.4.3) wie z. B. der Beschreibung geometrischer Objekte durch Parametergleichungen;
- Untersuchung interessanter, nichtlinearer geometrischer Objekte unter Zuhilfenahme von Computervisualisierungen – „Geometrisierung“ des Unterrichts.

### **Entwicklung heuristischer Strategien**

- Entwicklung der Fähigkeit der Schüler, sowohl gedanklich als auch visuell mithilfe des Computers durchgeführte Schnittbetrachtungen für die Lösung raumgeometrischer Probleme und Aufgaben zu nutzen und somit drei- auf zweidimensionale Sachverhalte zurückzuführen;
- Einbeziehung experimenteller Arbeitsweisen in den Unterricht, gezielter Einsatz von Versuch-und-Irrtum-Methoden für die Lösung von Positionierungsproblemen und die Modellierung geometrischer Formen;
- Förderung der Fähigkeit modularen Arbeitens (vgl. Abschnitt 1.5, S. 57).

### **Anwendungsbezug und Modellierung**

- Sichtbarmachen der Bedeutung analytischer Beschreibungen geometrischer Objekte für die Modellierung von Formen in der 3D-Computergrafik;
- Anwendung und Motivierung zentraler Gegenstände der analytischen Geometrie (wie des Skalarproduktes und von Normalenvektoren) aufgrund ihrer Bedeutung für die Bildberechnung in der 3D-Computergrafik; mathematische Modellierung physikalischer Vorgänge und ihrer Umsetzung in der Computergrafik.

### **Vertikale und horizontale Vernetzungen sowie fächerübergreifende Bezüge**

- Anknüpfen an in der Sekundarstufe I behandelte Inhalte der Elementargeometrie und Trigonometrie;
- Herstellen von Bezügen zur Analysis durch die Verwendung (nichtlinearer) elementarer Funktionen zur Beschreibung geometrischer Objekte und evtl. darüber hinaus Thematisierung der Darstellung durch Gleichungen beschriebener Objekte durch Grafiksoftware;<sup>18</sup>
- Herstellen fächerübergreifender Bezüge zu den Fächern Physik und Kunst;

---

<sup>18</sup>Letztgenanntes Ziel dürfte – wie generell die Herstellung von Bezügen zwischen der analytischen Geometrie des Raumes und der Infinitesimalrechnung – nur mit besonders leistungsstarken Kursen realisierbar sein (siehe S. 52f. und S. 298).

- Einbeziehung für die Informatik spezifischer Sicht- und Arbeitsweisen in den Unterricht der analytischen Geometrie.<sup>19</sup>

### **Förderung von Interesse und Motivation, Entwicklung der Unterrichtskultur**

- Motivierung der Schüler durch den ästhetischen Reiz der 3D-Computergrafik und die Möglichkeit, selbst dreidimensionale Szenen und Animationen zu generieren;
- Förderung der Kreativität der Schüler in dem auf S. 56 beschriebenen Sinne;
- Ermöglichen eigenständigen Arbeitens von Schülern oder Schülergruppen;
- Aufnahme projektbezogener Arbeitsformen in den Unterricht.

Nicht alle der genannten Ziele werden in jedem Falle verfolgt und erreicht werden können; die Möglichkeiten hierfür hängen von den zeitlichen Rahmenbedingungen und den Curricula sowie (insbesondere bei dem erwähnten Ziel der Herstellung von Bezügen zur Analysis) von den Voraussetzungen der Lerngruppe ab. Vorschläge für die Umsetzung der aufgeführten Ziele bei der Behandlung konkreter Themengebiete werden in Kapitel 4 unterbreitet.

### **3.2.2 Aspekte und Herangehensweisen**

Wie bereits in Abschnitt 1.6.2 ausgeführt wurde, bestehen zwei grundsätzliche *Aspekte der Einbeziehung der 3D-Computergrafik* in den Unterricht:

- (W) Dreidimensionale Grafiksoftware<sup>20</sup> ermöglicht die Visualisierung geometrischer Objekte und Relationen wie z. B. Lagebeziehungen. Sie eignet sich zudem für die Unterstützung experimentellen Arbeitens, u. a. indem Gleichungen aufgestellt und variiert und die entstehenden Flächen bzw. Körper betrachtet werden. Eine geeignete Software kann somit als „*Werkzeug*“ für den Unterricht dienen.
- (G) Wesentliche mathematische Grundlagen der 3D-Computergrafik können als Anwendungen des Lehrstoffes der analytischen Geometrie betrachtet werden und in einigen Fällen die Behandlung von Objekten und Verfahren der analytischen Geometrie motivieren. Elemente der 3D-Computergrafik bieten sich insofern als *Gegenstände des Unterrichts* an.<sup>21</sup>

---

<sup>19</sup>Genannt seien objektorientierte Sichtweisen (geometrische Objekte als Instanzen gewisser Klassen mit spezifischen Attributen und Methoden), modulares Arbeiten (s. o.) sowie iterative und rekursive Arbeitsweisen. Wie in Abschnitt 1.5 dargelegt wurde, handelt es sich dabei teilweise um Inhalte, die auch unabhängig von ihrer Bedeutung in der Informatik für den Mathematikunterricht relevant sind. Bezüge zur Informatik sind weiterhin hinsichtlich verschiedener Auffassungen des Vektorbegriffs von Interesse; darauf wird in Abschnitt 3.3.1 ab S. 194 noch näher eingegangen.

<sup>20</sup>Der Begriff „3D-Grafiksoftware“ wird bis zur Diskussion um die Auswahl geeigneter Software in Abschnitt 3.4 in einem allgemeinen Sinne verwendet und umfasst alle Programme, die dreidimensionale geometrische Objekte grafisch darstellen können, also z. B. auch CAS.

<sup>21</sup>Die Schüler erlangen dabei ein grundlegendes Verständnis für die Funktionsweise der Computergrafik. Im Sinne der auf S. 62 zitierten Äußerung von Schupp wird damit ein Beitrag zur Herausbildung von „Medienkompetenz“ geleistet.

Unter diesen beiden Aspekten ist der Werkzeugaspekt (W) als derjenige anzusehen, der weniger konzeptionelle Überlegungen erfordert. Er ist auch auf der Basis der bestehenden Curricula direkt umsetzbar.<sup>22</sup> Die Thematisierung von mathematischen Grundlagen der 3D-Computergrafik erfordert hingegen in stärkerem Maße Überlegungen zu ihrer Einbindung in einen Unterrichtsaufbau, dessen sachlogische Struktur sich nicht an den Grundlagen und der Wirkungsweise der Computergrafik ausrichten kann, sondern an fundamentalen Ideen der analytischen Geometrie orientieren muss (vgl. Abschnitt 1.4.3 sowie [244] und [247]).

Die (zumindest punktuelle) Behandlung von mathematischen Grundlagen der Computergrafik sollte nicht nur theoretisch erfolgen, sondern die Verwendung einer Grafiksoftware einschließen – die Umsetzung von Aspekt (G) setzt also (W) voraus. Dabei ist auf Grundlage der in Kapitel 2 beschriebenen Funktionsweise der 3D-Computergrafik eine sorgfältige Auswahl von Bezugspunkten zu praktischen Anwendungen vorzunehmen, um die Gefahr zu umgehen, dass theoretische Überlegungen bezugslos neben der Verwendung einer Software stehen. Wie bereits mehrfach erwähnt wurde, ist die motivierende Wirkung von Anwendungen mathematischen Wissens, die durch die Schüler nicht aktiv (nach)vollzogen werden können, begrenzt. Eine Konzeption für die Einbeziehung der 3D-Computergrafik in den Unterricht sollte daher so aufgebaut sein, dass sich theoretische Überlegungen und die praktische Arbeit mit computergrafischer Software gegenseitig anregen und befruchten. Dafür bieten sich u. a. folgende Unterrichtsszenarien an:

- Die Schüler entdecken durch visuelles Experimentieren „neue“ geometrische Objekte (z. B. als Schnittmengen), die sie anschließend mathematisch beschreiben. Durch theoretisch (anhand von Gleichungen) gewonnene Verallgemeinerungen modellieren sie weitere Formen, veranschaulichen und untersuchen diese.
- Durch die Arbeit mit der Software werden Fragen nach deren Funktionsweise aufgeworfen, die zunächst praktisch motiviert sind, z. B.: Wie können Videos erzeugt werden? Welche Parameter sind in welcher Weise zu setzen, um bestimmte optische Effekte auf Oberflächen zu erzielen? Davon ausgehend werden mathematische Modellbildungen erarbeitet und praktisch überprüft.

Konzepte für die Einbeziehung ausgewählter Gegenstandsbereiche der Computergrafik in den Unterricht, bei denen diese beiden Szenarien zur Anwendung kommen, werden in Kapitel 4 beschrieben. Da konkrete Überlegungen nicht in jedem Falle unabhängig von der verwendeten Software geführt werden können, ist diesbezüglich vorher eine Auswahl zu treffen. Zunächst erscheint jedoch eine Diskussion der Frage notwendig, welche Konsequenzen die Einbeziehung von Elementen der 3D-Computergrafik für die Behandlung zentraler Inhalte des Stoffgebietes Analytische Geometrie haben kann.

---

<sup>22</sup>Allerdings dürfte die Beschränkung auf die Visualisierung der in den gegenwärtigen Rahmenplänen hauptsächlich auftretenden linearen Objekte den Reiz und damit die motivierende Wirkung computergrafischer Visualisierungen erheblich beeinträchtigen.

### 3.3 Sichtweisen auf zentrale Gegenstände der analytischen Geometrie im Kontext der Einbeziehung von Elementen der Computergrafik

#### 3.3.1 Zur Einführung und Verwendung von Vektoren

Die Einbeziehung von Elementen der Computergrafik in den Unterricht der analytischen Geometrie führt zu der Notwendigkeit, Sichtweisen der Informatik hinsichtlich zentraler Begriffe und Inhalte des Stoffgebietes bei Überlegungen zu ihrer Behandlung im Unterricht zu berücksichtigen. Dies betrifft in besonderer Weise den Begriff des Vektors, der eine zentrale Bedeutung im gegenwärtigen Unterricht hat.<sup>23</sup> Eine einseitig an der Verwendung in der Informatik ausgerichtete Behandlung von Vektoren im Mathematikunterricht verbietet sich jedoch; vielmehr sind diesbezügliche didaktische Überlegungen zu beachten und Aspekte der Informatik damit in Verbindung zu bringen.

#### Didaktische Positionen und Erfahrungen hinsichtlich des Vektorbegriffs

Prinzipiell bestehen zwei Möglichkeiten, den Begriff des Vektors einzuführen:

- Nach einer axiomatischen Definition von Vektorräumen ist eine allgemeine Definition des Begriffs Vektor als Element eines Vektorraumes möglich.
- Vektoren können anhand konkreter Modelle der Vektorraumaxiome eingeführt werden, wobei vor allem die folgenden Modelle für den Unterricht relevant sind:
  - das arithmetische Modell der  $n$ -Tupel reeller Zahlen,
  - geometrische Modelle wie Punkte, „Zeiger“ (Ortsvektoren), Translationen sowie Pfeilklassen (vgl. [247], S. 159ff.).

Die Einführung von Vektoren als Elemente (axiomatisch definierter) Vektorräume ist im Mathematikunterricht nach den damit gesammelten Erfahrungen im Zusammenhang mit den Unterrichtsreformen der „Neuen Mathematik“ (vgl. Abschnitt 1.2.3) nicht mehr ernsthaft in Erwägung zu ziehen und wird in der didaktischen Diskussion als Grundlage für den Unterricht mittlerweile nahezu einhellig abgelehnt.<sup>24</sup> Vor- und Nachteile der genannten arithmetischen und geometrischen Modelle stellte TIETZE in [247] (S. 160ff.) gegenüber, wobei er feststellte, dass die Zugrundelegung jedes dieser Modelle Probleme aufweist.<sup>25</sup> Am stärksten durchsetzen konnten sich in deutschsprachigen Schulbüchern

---

<sup>23</sup>Analoge Konsequenzen ergeben sich für die Behandlung von Matrizen, falls diese erfolgt.

<sup>24</sup>Allerdings wirken Relikte einer im Zuge der „Neuen Mathematik“ in den Unterricht gelangten strukturorientierten linearen Algebra noch heute insofern nach, als zumindest die Gliederungen der Lehrgänge zur analytischen Geometrie nach vielen Rahmenplänen (siehe 1.3) an Strukturüberlegungen ausgerichtet sind und sich die betrachteten geometrischen Objekte diesen unterordnen.

<sup>25</sup>So führt die auf den ersten Blick aufgrund ihrer Einfachheit interessante Interpretation von Vektoren als Punkte zu erheblichen Schwierigkeiten bei der Addition sowie hinsichtlich der Länge und der Orthogonalität von Vektoren. Die Abhängigkeit von einem Koordinatenursprung widerspricht zudem einer grundlegenden Eigenschaft des Vektorbegriffs. FÜHRER schlug diesbezüglich vor, affine Objekte wie Punkte, Geraden und Ebenen als „bis auf Parallelverschiebung linear“ aufzufassen, wodurch eine

das Modell der Pfeilklassen und (meist im Zusammenhang damit, nämlich als Interpretation von Pfeilklassen) das der Verschiebungen. Auf eine explizite Erörterung des Begriffs der Äquivalenzklasse (welcher dem Pfeilklassenmodell zugrunde liegt) wird dabei zunehmend verzichtet. Stattdessen lassen sich in neueren Schulbüchern hauptsächlich Einführungen des Vektorbegriffs finden, bei denen eine Veranschaulichung von Pfeilklassen unter Bezugnahme auf Translationen erfolgt. Als Beispiel sei die einführende Beschreibung des Begriffs „Vektor“ in dem Lehrbuch von 2004 für Leistungskurse aus der Reihe Lambacher Schweizer angegeben:

„**Vektoren:** In der Geometrie meint man mit der Bezeichnung Vektor den Spezialfall einer Verschiebung; deshalb wird ein Vektor in der Geometrie durch eine Menge zueinander paralleler, gleich langer und gleich gerichteter Pfeile beschrieben.“ ([8], S. 32)

Es deutet sich hierbei und bei ähnlichen Formulierungen ein sprachlich-inhaltliches Problem an. Die Einführung des Begriffs der Äquivalenzklasse soll vermieden werden; die Gleichsetzung von Vektoren mit einzelnen Pfeilen wird jedoch dem Vektorbegriff nicht gerecht, so dass eine „Umschreibung“ des Klassenkonzepts erforderlich ist, um Vektoren angemessen durch Pfeile beschreiben zu können. Die Auffassung von Vektoren als Mengen von Pfeilen stellt zudem ein Problem bei der Einführung der Rechenoperationen mit Vektoren dar, da hierbei mit Repräsentanten gearbeitet werden muss.

Aufgrund der angedeuteten Probleme mit der Einführung von Vektoren anhand geometrischer Modelle wurden Vorschläge unterbreitet, den Vektorbegriff „arithmetisch“ anhand von  $n$ -Tupeln reeller Zahlen zu begründen. In Kursen zur linearen Algebra, in denen nichtgeometrische Anwendungen einen wichtigen Schwerpunkt darstellen, liegt diese Herangehensweise ohnehin nahe. So führten PROFKE et al. in dem Schulbuch [97] Vektoren als  $n$ -Tupel ein und deuteten sie (nach der Behandlung nichtgeometrischer Anwendungen) geometrisch. Ähnlich gingen ARTMANN und TÖRNER in [4] vor, die im Zusammenhang mit linearen Gleichungssystemen gleichzeitig Vektoren und Matrizen definierten. Direkter auf die geometrische Anwendung arithmetisch eingeführter Vektoren zielten BÜRGER, FISCHER, MALLE und REICHEL in ihrem Beitrag [36] und einem – dem dort herausgearbeiteten Konzept „*Arithmetische Vektoren mit geometrischer Deutung*“ folgenden – Schulbuch. Neben der vergleichsweise einfachen Definition von Vektoren als  $n$ -Tupel (speziell Paare und Tripel) und der ebenfalls einfachen Einführung des Rechnens mit Vektoren nannten sie folgende Vorteile dieses Konzepts:

- *Freie Beweglichkeit zwischen verschiedenen Deutungen des Vektorbegriffs*

Arithmetische Vektoren lassen sich in verschiedener Weise geometrisch interpretieren; umgekehrt können z. B. Pfeile oder Punkte arithmetisch erfasst werden. Der Terminus Vektor bleibt dabei an die  $n$ -Tupel gebunden und ermöglicht als Bindeglied den Übergang zwischen Pfeil- und Punktdeutungen (siehe [36], S. 172f.).

---

Identifikation von Punkten und Vektoren möglich wird ([85], 1979, S. 33f.). Allerdings erscheint auch eine derartige Vorgehensweise aufgrund der damit verbundenen stark abstrahierten Auffassung den Schülern vertrauter Objekte, die an eine Lage im Raum gebunden sind, fragwürdig. In der jüngeren Diskussion wurden keine in diese Richtung zielenden Vorschläge mehr unterbreitet.

- *Möglichkeit der Übertragung von Konzepten zwischen  $\mathbb{R}^n$  und der Geometrie*

BÜRGER et al. wiesen darauf hin, dass eine solche Übertragung von Konzepten sowie Sprechweisen einer sprachlichen Vereinfachung und einer „Geometrisierung der Sprache“ dienen kann, die von heuristischem Wert ist ([36], S. 177f.).<sup>26</sup> Dennoch ist es möglich und meist sinnvoll, für arithmetische Vektoren je nach geometrischer Interpretation unterschiedliche Symbole zu verwenden, z. B.  $A$  bei der Punkt- und  $\vec{a}$  bei der Pfeil-Interpretation.

TIETZE hob als weitere Vorteile der Nutzung des  $n$ -Tupel-Modells als primäres Vektormodell hervor, dass sich die Struktur der  $n$ -Tupel ausgehend von den Fällen  $n = 2, 3$  gut verallgemeinern lässt und ein zentrales Mathematisierungsmuster darstellt. Insbesondere können  $n$ -Tupel als Bindeglied zwischen geometrischen Inhalten und Themen einer anwendungsorientierten linearen Algebra fungieren ([247], S. 164). Aus Sicht der Thematik der vorliegenden Arbeit ist hinzuzufügen, dass die Nutzung eines arithmetischen Vektormodells eine Verbindung von Vektorkonzepten der analytischen Geometrie und der Informatik herstellt, worauf noch eingegangen wird.

So fundamental die Einführung des Vektorbegriffs für den Aufbau von Lehrgängen einer vektoriell orientierten analytischen Geometrie aus sachlogischer Sicht ist, so interessant sind andererseits Ergebnisse empirischer Untersuchungen von WITTMANN, aus denen hervorgeht, dass die Art der Einführung des Vektorbegriffs und das dazu herangezogene Modell einen geringen Einfluss auf die individuellen Konzepte und Vorstellungen haben, die Schüler hinsichtlich von Vektoren aufbauen ([276], S. 110–123, 371–376). Als bestimmend hierfür erweisen sich in erheblich stärkerem Maße die *Kontexte, innerhalb derer Schüler im Verlauf des Unterrichts mit Vektoren arbeiten*. Für Schüler von Kursen, in denen der Vektorbegriff arithmetisch eingeführt und geometrisch interpretiert sowie im weiteren Unterricht für Gegenstände der analytischen Geometrie (Parametergleichungen von Geraden und Ebenen, Skalarprodukt, Winkel- und Abstandsberechnungen, Normalengleichungen) angewendet wurde, konstatierte WITTMANN ([276], S. 374):

*„... eine Gemeinsamkeit wird überdeutlich: die Neigung von Schülern, Begriffe wie ‚Vektor‘, ‚Betrag eines Vektors‘ oder ‚Skalarprodukt zweier Vektoren‘ eher an konkret-gegenständliche geometrische Objekte und damit verbundene Operationen zu knüpfen als an  $n$ -Tupel oder Zahlen und damit verbundene Operationen. In den impliziten Teilkonzepten (der Schüler) besitzen  $n$ -Tupel nicht den Status eigenständiger Objekte, sondern dienen lediglich der gemeinsamen, verbindenden Beschreibung von Punkten und Pfeilen im Koordinatensystem. Verglichen mit dem intendierten arithmetischen Vektorbegriff stellt dies eine Umkehrung der Ontologie dar.“*

Dieses Ergebnis bezieht sich allerdings, wie bereits erwähnt wurde, auf einen Unterrichtsaufbau, bei dem Vektoren zunächst, gewissermaßen „vorbereitend“, arithmetisch eingeführt und im Folgenden fast ausschließlich in geometrischen Kontexten verwendet werden. Die Zugrundelegung eines arithmetischen Vektorbegriffs sieht WITTMANN

---

<sup>26</sup>Die Autoren betonten, dass sprachliche Identifikationen z. B. der Elemente von  $\mathbb{R}^n$  mit Punkten sowie von Teilmengen mit Geraden und Ebenen keine Verfälschungen der jeweiligen Situation bedeuten müssen und derartige Identifikationen im Eindimensionalen allgemein üblich sind ([36], S. 179).

dann als sinnvoll an, wenn dieser in größerem Maße in nichtgeometrischen Kontexten verwendet wird. Dies ist in Kursen der anwendungsorientierten linearen Algebra der Fall, kann allerdings auch, wie im Folgenden erörtert wird, aus der Einbeziehung von Elementen der Computergrafik in einen geometrisch orientierten Lehrgang resultieren.

### Der Vektorbegriff in der Informatik und speziell in der Computergrafik

Vektoren und Matrizen sind aus der Sicht der Informatik spezielle Felder (Arrays), bei denen es sich um Zusammenfassungen ein- oder mehrdimensional angeordneter Daten gleichen Typs handelt.<sup>27</sup> Bei der Verwendung von Programmiersprachen können die zu Arrays zusammengefassten Daten beliebigen Typs sein; es kann sich dabei u. a. um ganze (`int`, `long`) oder reelle Zahlen (`real` bzw. `float`),<sup>28</sup> Wahrheitswerte (`boolean`), Textzeichen oder beliebige Zeichenketten (`string`) handeln. Auch aus Arrays selbst können komplexere Arrays konstruiert werden. Somit ist es möglich, aus mehreren Vektoren (eindimensionalen Arrays)<sup>29</sup> eine Matrix zu erzeugen, die jedoch nur dann der in der Algebra üblichen Auffassung von Matrizen entspricht, wenn die verwendeten eindimensionalen Arrays jeweils gleich lang sind.<sup>30</sup> In gebräuchlichen Programmiersprachen wie C bzw. C++ und Java ist es möglich, Arrays von vornherein mehrdimensional zu initialisieren, so dass Matrizen auch direkt angelegt werden können. Die folgenden

Zeilen erzeugen den Vektor  $\vec{v} = \begin{pmatrix} 0,5 \\ 1 \\ -1,5 \end{pmatrix}$  und die Matrix  $M = \begin{pmatrix} 0,5 & 1 & -1,5 \\ -1,5 & 2 & 2,5 \\ 3 & 4 & 0,1 \end{pmatrix}$  in Java:

```
double [] v = {0.5, 1, -1.5};
```

```
double [][] M = { {0.5, 1, -1.5} , {-1.5, 2, 2.5} , {3, 4, 0.1} };
```

Generell kann auf Elemente von Arrays über Indizes zugegriffen werden,<sup>31</sup> bei den als Beispiele angegebenen Arrays mittels `v[i]` bzw. `M[i][j]` ( $0 \leq i, j \leq 2$ ):

```
v[1] → 1.0    und    M[0][2] → -1.5.
```

<sup>27</sup>Eine Verallgemeinerung von Arrays sind *Listen*, die unterschiedliche Datentypen enthalten können.

<sup>28</sup>Die Bezeichnung `real` bzw. reelle Zahl entspricht nicht exakt dem mathematischen Inhalt dieses Begriffs, da aufgrund der Endlichkeit des Speichers nur rationale Zahlen erfasst werden. Eine andere Bezeichnung lautet „Fließkommazahl“, wobei die Bezeichnung `float` oft für Fließkommazahlen verwendet wird, die 32 Bit an Speicher belegen, während bei Zahlen des Typs `double` durch die Nutzung von 64 Bit eine höhere Genauigkeit möglich ist.

<sup>29</sup>Der Begriff des Vektors wird hier in einem eingeschränkten Sinne verwendet, der in der Informatik gebräuchlich ist und sich an das  $n$ -Tupel-Modell anlehnt. In mathematischem Sinne bilden auch die Menge aller  $m \times n$ -Matrizen und die Menge aller Arrays reeller Zahlen (mit festgelegten Dimensionen der Unterarrays) bezüglich der Addition und der Multiplikation mit Skalaren Vektorräume.

<sup>30</sup>Ein Farbbild mit  $n$  Pixeln horizontaler und  $m$  Pixeln vertikaler Auflösung besitzt die Struktur eines dreidimensionalen Arrays von Zahlen mit  $n \cdot m \cdot 3$  Elementen, da die Farbe jedes Pixels durch drei Werte (in dem hauptsächlich verwendeten RGB-Modell für die Komponenten rot, grün und blau) beschrieben wird, vgl. 2.1.1 und 2.1.2. Es kann jedoch auch als zweidimensionales Array (Matrix) eindimensionaler Arrays (Vektoren) mit jeweils 3 Elementen angesehen werden.

<sup>31</sup>Standard-Arrays werden durch ganzzahlige Indizes gekennzeichnet. Dem jeweils ersten Element wird i. Allg. der Index 0 zugeordnet, was mitunter zu Verwirrungen führt, da Indizierungen von Vektor- oder Matrixelementen in der Mathematik meist mit 1 beginnen.



In *Computeralgebrasystemen* werden Vektoren grundsätzlich mit  $n$ -Tupeln identifiziert. Für deren Eingabe verwendet Derive Masken, in denen der Anwender entsprechende Felder ausfüllt. In MathCad, Mathematica und MuPAD entspricht die Eingabe von Vektoren der spezieller (einspaltiger) Matrizen; in MuPAD muss der Befehl `matrix` explizit verwendet werden, für den oben betrachteten Vektor  $\vec{v}$  z. B. in der Form

```
v := matrix([0.5, 1, -1.5]).
```

Mit derartig definierten Vektoren können die Vektoroperationen (einschließlich Skalar- und Vektorprodukt) ausgeführt werden; sehr leicht ist für die Dimensionen 2 und 3 auch eine Darstellung durch Pfeile (mit Anfangspunkt im Koordinatenursprung) möglich.

Die Szenenbeschreibungssprache des Grafikprogramms *POV-Ray* verwendet Vektoren und Matrizen in speziellerer Weise als Programmiersprachen und auch als CAS. Matrizen kommen ausschließlich für die Beschreibung geometrischer Transformationen zur Anwendung,<sup>32</sup> ihre Verwendung als übergreifende Datenstruktur ist nicht vorgesehen. Vektoren treten in mehreren Kontexten auf, wobei  $n$ -Tupel (hauptsächlich mit  $n = 3$ , in einigen Zusammenhängen aber auch mit  $n = 2, 4$  oder  $5$ ) verschiedene Objekte und Eigenschaften beschreiben; die wichtigsten davon seien hier zusammengefasst:

- *Punkte des Raumes* werden durch Koordinatentripel beschrieben.
- Für die *Beschreibung spezieller geometrischer Transformationen* werden ebenfalls Tripel genutzt: `translate`  $\langle t_x, t_y, t_z \rangle$ , `rotate`  $\langle \varphi_x, \varphi_y, \varphi_z \rangle$ , `scale`  $\langle s_x, s_y, s_z \rangle$ . Dabei ist die vektorielle Beschreibung von Verschiebungen mit dem in der Schulmathematik gebräuchlichen Modell der Verschiebungsvektoren vereinbar, während die Beschreibungen von Drehungen (durch die EULERSchen Winkel) und Skalierungen (durch die Streckungsfaktoren in Richtungen der drei Koordinatenachsen) mit dem Vektorbegriff nur über das  $n$ -Tupel-Modell im Zusammenhang stehen.
- *Farben* werden durch RGB-Vektoren dargestellt:<sup>33</sup> `color rgb`  $\langle r, g, b \rangle$ . Durch die Erweiterung auf Quadrupel bzw. Quintupel wird dieses Modell um die Beschreibung zweier Arten von Transparenz erweitert:<sup>34</sup>  
`color rgbf`  $\langle r, g, b, f \rangle$ , `color rgbt`  $\langle r, g, b, t \rangle$ , `color rgbft`  $\langle r, g, b, f, t \rangle$ .

Unabhängig davon, in welchem Kontext Vektoren in *POV-Ray* auftreten, wird immer dieselbe Syntax  $\langle x_1, \dots, x_n \rangle$  verwendet. Die gebräuchlichen Vektoroperationen stehen auch in *POV-Ray* zur Verfügung.

Zusammengefasst liegt der Vektorauffassung in der Informatik und speziell in der Computergrafik das  $n$ -Tupel-Modell zugrunde. In der *POV-Ray*-Dokumentation wird dies folgendermaßen zum Ausdruck gebracht:

„A vector is a set of related float values.“

<sup>32</sup>Beispiele für die Verwendung von Transformationsmatrizen in *POV-Ray* enthält Abschnitt 2.4.

<sup>33</sup>Siehe dazu Abschnitt 2.1.2; wie dort bereits erwähnt wurde, ist die Bezeichnung „Vektor“ hierfür nicht korrekt, da die R-, G- und B-Werte nicht über ganz  $\mathbb{R}$  definiert sind, aber dennoch gebräuchlich.

<sup>34</sup>Dabei steht **f** für gefilterte und **t** für rein transmittierende Transparenz, siehe Abschnitt 2.5.1.

## Zusammenfassung und Schlussfolgerungen

Die vorangegangenen Überlegungen haben deutlich werden lassen, dass in der Informatik und auch speziell in der Computergrafik Vektoren in arithmetischem Sinne als  $n$ -Tupel aufgefasst werden und sehr unterschiedliche Objekte und Eigenschaften – von Punkten bis hin zu Farben – beschreiben. Diese Tatsache legt die Verwendung eines arithmetischen Vektormodells nahe, wenn im Unterricht Elemente der Computergrafik in größerem Umfang thematisiert werden und die Schüler mit einer entsprechenden Software arbeiten. Der Ansatz von BÜRGER et al., Vektoren arithmetisch einzuführen und geometrisch zu deuten (vgl. S. 192f.), wird auch dem in der Computergrafik verwendeten Vektorbegriff gerecht. Das erste Vektormodell, das die Schüler kennenlernen, wenn sie mit einer 3D-Grafiksoftware arbeiten, ist das der  $n$ -Tupel. Dies wird jedoch nicht zwangsläufig zu der Herausbildung eines Verständnisses für den Begriff des Vektors und seine Leistungsfähigkeit führen, sondern lediglich zu der Erkenntnis, dass durch  $n$ -Tupel (hauptsächlich Tripel) recht unterschiedliche „Dinge“ beschrieben werden können.

Das Aufgreifen der RGB-Beschreibung von Farben<sup>35</sup> im Unterricht kann m. E. einen Beitrag dazu leisten, dass die Schüler Vektoren (im Sinne von Zahlentripeln) als Bindeglied zwischen scheinbar zusammenhanglosen nichtgeometrischen und geometrischen Objekten erkennen. Die Darstellung von Farben durch RGB-Tripel und die anschließende Interpretation dieser „Farbvektoren“<sup>36</sup> als Ortsvektoren ermöglicht die sinnvolle geometrische Veranschaulichung eines zunächst nichtgeometrischen Inhalts – sie führt zum Farbwürfel (RGB-Würfel, siehe S. 77 und Abb. B.3 auf S. 349). Die FarbadDITION (einschließlich entsprechender Experimente in einer Bildbearbeitungssoftware und der Herstellung von Bezügen zu den Punkten des Farbwürfels) stellt eine interessante Ergänzung zu arithmetischen und geometrischen Interpretationen der Vektoraddition dar.<sup>37</sup> Dennoch stellt die Beschreibung von Farben nur *eine* nichtgeometrische Anwendung von Vektoren dar, die zudem im Unterricht der analytischen Geometrie nur in Ansätzen aufgegriffen werden kann. Geometrische Kontexte der Verwendung von Vektoren werden hier weiterhin dominieren.

---

<sup>35</sup>Da die Beschreibung von Farben durch RGB-Tripel in praktisch jedem Bildbearbeitungs- und Grafikprogramm sowie bei der Gestaltung von Internetseiten auftritt, ist es recht wahrscheinlich, dass in den meisten Kursen jeweils einige Schüler damit bereits Erfahrungen gemacht haben. Diese können z. B. durch Schülervorträge in den Unterricht eingebracht werden und die Sicht der Schüler auf den Begriff des Vektors erweitern.

<sup>36</sup>Wie bereits erwähnt wurde, dürften allerdings „Farbvektoren“ in strengem Sinne nicht als Vektoren bezeichnet werden, da die Komponenten jeweils nur Werte von 0 bis 1 bzw. ganzzahlige Werte von 0 bis 255 annehmen können (siehe S. 76). Dieser Unterschied zu anderen Beispielen für Vektoren sollte den Schülern deutlich werden; er lässt sich anhand des Farbwürfels gut veranschaulichen.

<sup>37</sup>Weitergeführt werden können die Überlegungen zu Farbvektoren anhand weiterer Farbmodelle. Während die Zusammenhänge zum CMY-Modell (siehe S. 77f.) recht elementar, aber m. E. weniger interessant sind, treten bei der Betrachtung des HSB-Modells (S. 78f.) geometrische Überlegungen auf, die zu dem in der Kunst interessanten Farbkreis führen. Hierbei bieten sich gute Ansatzpunkte für eine fächerübergreifende Kooperation. Allerdings wird im Mathematikunterricht oft die Zeit fehlen, die für über das RGB-Modell hinausgehende Betrachtungen von Farbmodellen nötig wäre.

Grundsätzlich wird m. E. keine vollständig zufrieden stellende Antwort auf die Frage nach dem für die Einführung von Vektoren im Mathematikunterricht am besten geeigneten Modell zu geben sein. Der Vektorbegriff ist ein weit tragender Begriff, der eine Abstraktion sehr verschiedener Objekte und Zusammenhänge ermöglicht. Sein übergreifendes Verständnis setzt – wie bereits im Zusammenhang mit seiner historischen Entwicklung deutlich wurde, vgl. Abschnitt 1.1 – eine breite Abstraktionsbasis voraus, über die nur wenige Schüler verfügen. Jede Einführung des Vektorbegriffs in der Schule wird daher mit Kompromissen und Nachteilen verbunden sein. Allerdings ist die Frage, wie Vektoren *eingeführt* werden, nach den bereits erwähnten Untersuchungen von WITTMANN für die bei den Schülern herausgebildeten Auffassungen dieses Begriffs von untergeordneter Bedeutung gegenüber den Kontexten, in denen mit Vektoren *gearbeitet* wird (siehe S. 193). Die Relevanz der Frage nach dem primär verwendeten Vektormodell relativiert sich dadurch gegenüber der Bearbeitung geeigneter Aufgaben, anhand derer verschiedene Aspekte des Vektorbegriffs deutlich werden, sowie reflektierenden Unterrichtsphasen, nachdem die Schüler bereits mit Vektoren gearbeitet haben.

Aus den aufgeführten Überlegungen heraus erscheinen m. E. folgende Schlussfolgerungen für den Umgang mit Vektoren im Unterricht sinnvoll:

- Die *Einführung von Vektoren* sollte *nicht am Anfang des Stoffgebietes* Analytische Geometrie stehen. Die Schüler dürften anhand der Beschreibung von Punkten und einfachen geometrischen Objekten durch Koordinaten und (zumindest in ersten Ansätzen) durch Gleichungen in Koordinaten einen anschaulicheren und einfacheren Einstieg in die analytische Geometrie finden,<sup>38</sup> als wenn gleich zu Anfang der vergleichsweise abstrakte Vektorbegriff gewissermaßen „auf Vorrat“ eingeführt wird. Zudem führt die Überbetonung des Vektorbegriffs und die Ausrichtung des gesamten Stoffgebietes an der Verwendung vektorieller Vorgehensweisen zu einer geometrischen Verarmung (vgl. 1.4.5).<sup>39</sup>
- Die Einführung des Vektorbegriffs sollte in *Abstraktion vorher betrachteter Beispiele* erfolgen. Dazu gehören neben den aus der SI bzw. dem Physikunterricht bekannten Verschiebungen, Kräften und Geschwindigkeiten vor allem Koordinaten von Punkten, „Richtungspfeile“ (mit denen bereits vor der Einführung des Vektorbegriffs gearbeitet werden kann) sowie durch RGB-Tripel beschriebene Farben.<sup>40</sup>

---

<sup>38</sup>Ein derartiger, mit der Modellierung geometrischer Formen in einer Grafiksoftware verbundener Einstieg ist Gegenstand von Abschnitt 4.1.

<sup>39</sup>In Curricula von Grundkursen mit recht wenig zur Verfügung stehender Zeit ist auch der Verzicht auf die Behandlung von Vektoren in Erwägung zu ziehen. Die dadurch gewonnene Zeit kann für eine geometrisch interessante Koordinatengeometrie m. E. ertragreicher genutzt werden als für die Einführung des Vektorbegriffs. Wie die Untersuchungen von WITTMANN (siehe [276] und Beiträge in [247]) zeigen, verbindet die überwiegende Mehrheit der Schüler mit diesem Begriff zum Abschluss des Stoffgebietes nur vage und an spezielle Kontexte gebundene Vorstellungen.

<sup>40</sup>Das Beispiel der Farben ist dann von Interesse, falls Aspekte der Computergrafik im Unterricht thematisiert werden. In anderen Fällen bieten sich andere nichtgeometrische Beispiele wie Stücklisten an (siehe z. B. [97]). Falls eine große Zahl von Schülern eines Kurses den Informatikunterricht besucht, liegt natürlich auch ein Bezug zur Auffassung von Vektoren als eindimensionale Arrays nahe.

- Da  $n$ -Tupel als „Bindeglied“ zwischen allen betrachteten Vektorraummodellen fungieren können und zudem immer auftreten, wenn Vektoren zu Zwecken der grafischen Darstellung oder für Berechnungen in den Computer eingegeben werden, erscheint das  $n$ -Tupel-Modell als übergreifendes Modell am geeignetsten. Es muss aber angemerkt werden, dass bei den Schülern damit noch keine Auffassung von  $n$ -Tupeln als „eigenständigen“ Objekten entstehen muss, sondern sie diese vielmehr als universelle Möglichkeit der *Beschreibung* unterschiedlicher Objekte und Eigenschaften sowie des *Operierens* damit betrachten werden. Gelingt dies, so haben die Schüler einen Einblick in die Leistungsfähigkeit und Universalität des Vektorbegriffs im Sinne der mathematischen Modellierung erhalten. Ein strukturelles Verständnis des Vektorbegriffs ist nach allen Erfahrungen bei Schülern nur in Ausnahmefällen zu erreichen – dies gilt sowohl im Hinblick auf die mathematische Struktur des Vektorraumes als auch für Vektoren als Äquivalenzklassen.<sup>41</sup>
- *Pfeile sind als Repräsentanten von Vektoren und für deren Veranschaulichung* in jedem Falle – unabhängig von der Wahl des primären Modells für die Einführung des Vektorbegriffs – von hoher Bedeutung. Dies hat mindestens zwei Ursachen:
  - Mit dem Begriff des Vektors wird die Mehrheit der Schüler erstmalig im Physikunterricht der SI konfrontiert. Sie lernen dort die Kraft als „vektorielle Größe“ kennen, worunter eine Größe verstanden wird, die einen Betrag sowie eine Richtung und einen Richtungssinn besitzt. Die Addition von Kräften erfolgt im Kräfteparallelogramm mithilfe von Pfeilen. Diese Vorerfahrung der Schüler sollte nicht ignoriert werden.<sup>42</sup>
  - Zu den ersten und wichtigsten Anwendungen von Vektoren im Stoffgebiet Analytische Geometrie gehören Parameterdarstellungen von Geraden und Ebenen. Die Interpretation der Richtungsvektoren als Pfeile liegt nahe und ist für die Veranschaulichung von Parameterdarstellungen sinnvoll. Ähnliches gilt bei der Behandlung von Winkeln sowie von Normalengleichungen.

Die Interpretation durch Pfeile wird aus diesen Gründen die Schülerkonzepte von Vektoren in einem Unterricht der vektoriellen analytischen Geometrie maßgeblich mitbestimmen. Die Einbeziehung von Elementen der Computergrafik ändert hieran nichts, denn obwohl sie eine Arbeit mit  $n$ -Tupeln erfordert, sind Pfeile von Bedeutung für die Veranschaulichung von Vektoren in CAS oder Grafiksoftware.

---

<sup>41</sup>In Kursen, die aus besonders leistungsstarken Schülern bestehen, können nach der Behandlung verschiedener Vektormodelle im Sinne einer abstrahierenden Axiomatik die Vektorraumaxiome als Eigenschaften und Rechenregeln herausgearbeitet werden. Die Schüler sollten erkennen, dass mit den Axiomen die strukturell wesentlichen Eigenschaften der behandelten Modelle erfasst sind. Auf diese Weise ein Verständnis des Vektorraumes als mathematische Struktur zu erreichen, wird jedoch m. E. nur in Ausnahmefällen, z. B. in mathematischen Spezialklassen, möglich sein.

<sup>42</sup>Nach meinen Erfahrungen verankert sich die im Physikunterricht der SI verwendete, an Kräfte gebundene Vektorauffassung bei vielen Schülern recht stark und ist im 13. Schuljahr noch präsent. Vergleichsweise selten werden von den Schülern Geschwindigkeiten als Beispiele für Vektoren genannt, obwohl nach vielen Physik-Rahmenplänen in Klasse 11 die Geschwindigkeit vektoriell behandelt wird. Da sich diese Erfahrungen auf einzelne Kurse beziehen, ist ihre Verallgemeinerbarkeit jedoch fragwürdig.

- Der *Vektorbegriff* sollte in einem Zusammenhang eingeführt werden, in dem damit eine *Vereinfachung* von Beschreibungen erreicht werden kann. Eine Gelegenheit hierfür stellt die Behandlung der Parameterdarstellungen von Geraden dar, die nach einer Unterrichtsreihe zur Koordinatengeometrie erfolgen könnte.<sup>43</sup>

Als Konsequenz aus diesen Schlussfolgerungen liegt es nahe, an die Behandlung des Vektorbegriffs und die Wahl des primären Modells, anhand dessen Vektoren eingeführt werden, pragmatisch heranzugehen und Vektoren – orientiert an dem jeweiligen Kontext ihrer Nutzung – flexibel zu verwenden. Dabei sind sowohl das  $n$ -Tupel-Modell als auch die Veranschaulichung durch Pfeile bedeutsam. Nach den erwähnten Untersuchungen und Erfahrungen erscheint es weder realistisch, bei einer großen Zahl von Schülern ein Verständnis für die Struktur des Vektorraumes zu erreichen, noch dürfte dies hinsichtlich des Verständnisses von Pfeilklassen als Äquivalenzklassen<sup>44</sup> möglich sein.

Eine strukturelle Überlegung, die m. E. notwendig ist, damit die Verwendung des Begriffs „Vektor“ als übergreifende Bezeichnung unterschiedlicher Objekte überhaupt sinnvoll erscheinen kann, betrifft die *Isomorphie* der betrachteten Vektor(raum)modelle, insbesondere des Pfeil(klassen)- und des  $n$ -Tupel-Modells. Derartige Überlegungen lassen sich (ohne explizite Verwendung des Begriffs der Isomorphie) elementar führen, indem die Schüler Pfeile grafisch addieren und vervielfachen, wobei auf Vorkenntnisse zur Addition von Kräften aus dem Physikunterricht der SI zurückgegriffen werden kann. Die Ergebnisse können sie mit denen entsprechender rechnerischer Operationen mit  $n$ -Tupeln vergleichen (Abb. 3.1). In der Ebene ist dies durch Zeichnungen auf Papier, im Raum vor allem mithilfe des Computers möglich (siehe dazu auch Abschnitt 4.2). Dabei lässt sich auch die Unabhängigkeit von der Lage bezüglich des Koordinatensystems thematisieren.

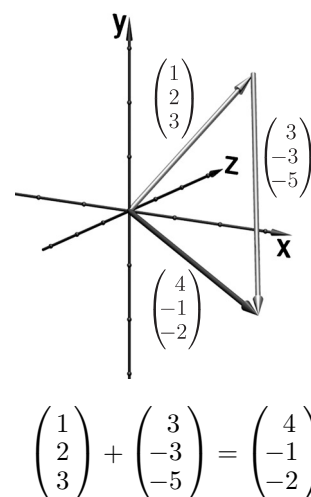


Abbildung 3.1: Addition von Vektoren

Zusammenfassend ist m. E. die folgende Begriffsbestimmung von Vektoren, die bei deren Einführung herausgearbeitet werden kann, eine sinnvolle Grundlage für die weitere

<sup>43</sup>In Abhängigkeit von der zur Verfügung stehenden Zeit ist daran zu denken, zunächst Parameterdarstellungen von durch zwei Punkte gegebenen Geraden ohne die Erwähnung von Vektoren zu behandeln und dann den Richtungsvektor zu thematisieren, so dass sichtbar wird, dass sich durch die vektorielle Beschreibung tatsächlich eine Vereinfachung ergibt. Oft ist es aus Zeitgründen aber notwendig, sofort vektoriell vorzugehen.

<sup>44</sup>Damit ist jedoch nicht gemeint, dass die Schüler Vektoren mit einzelnen Pfeilen identifizieren sollten; eine solche Gleichsetzung ist unbedingt zu vermeiden und zu verdeutlichen, dass die Bedeutung eines Vektors allgemeiner ist als die Darstellung durch einen konkreten Pfeil. Dazu können die Schüler die Pfeildarstellungen eines Vektors an verschiedene Punkte antragen – im Zweidimensionalen durch Zeichnungen auf Papier und im Raum mittels geeigneter mechanischer Modelle (sofern vorhanden) sowie einer geeigneten Software, siehe Abschnitt 4.2, S. 235. Es ist herauszuarbeiten, dass die dargestellten Pfeile jeweils ein- und denselben Vektor repräsentieren.

Arbeit im Unterricht, wobei später evtl. Exaktifizierungen und Verallgemeinerungen (beispielsweise durch den Übergang zu höheren Dimensionen) erfolgen können.

*Vektoren werden durch Paare oder Tripel reeller Zahlen dargestellt:*

$$\vec{a} = \begin{pmatrix} x_a \\ y_a \end{pmatrix} \text{ bzw. } \vec{a} = \begin{pmatrix} x_a \\ y_a \\ z_a \end{pmatrix}.$$

- *Vektoren können u. a. Punkte, Kräfte, Verschiebungen und Farben beschreiben.*<sup>45</sup>
- *Sie können in der Ebene oder im Raum durch Pfeile dargestellt werden, wobei gleich lange, parallele und gleich gerichtete Pfeile denselben Vektor darstellen.*

Nicht völlig vermeiden lässt sich bei dieser Begriffsauffassung die Unterscheidung zwischen Ortsvektoren und „freien“ Vektoren (die u. a. als Richtungsvektoren auftreten).<sup>46</sup>

### 3.3.2 Sichtweisen auf Parameterdarstellungen

Parameterdarstellungen von Geraden und Ebenen gehören zu den Standardinhalten aller an der Behandlung einer vektoriellen Raumgeometrie orientierten Curricula der analytischen Geometrie (vgl. Abschnitt 1.3). In den meisten Fällen folgen dabei nach einer Einführung der Parameterdarstellungen sehr schnell Aufgaben zur Umformung von Parameter- in Koordinatenform und umgekehrt sowie zur Untersuchung von Lagebeziehungen, der Bestimmung von Schnittpunkten sowie (meist etwas später) zu Abstands- und Winkelberechnungen (vgl. 1.4.2). Zwei wichtige – miteinander verbundene – Aspekte der analytischen Geometrie, die anhand der Parameterdarstellungen gut verfolgt werden könnten, kommen dabei nicht in ausreichendem Maße zur Geltung:

- Die Schüler gelangen höchstens in Ansätzen zu einer *Auffassung geometrischer Objekte als Punktmengen*, dominierend bleibt die Auffassung von Geraden als ganzheitlichen, konkret-gegenständlichen Objekten, deren Lage im Raum durch einen Punkt und einen Richtungsvektor festgelegt ist.<sup>47</sup>
- Der *funktionale Zusammenhang zwischen dem Parameter* (bzw. den Parametern) *und den zugehörigen Punkten* wird von den Schülern meist nicht erkannt. Das

<sup>45</sup>Hier sollten von den Schülern bereits bekannte Beispiele zusammengetragen werden, die sie mit dem Begriff des Vektors verbinden. Kräfte und Verschiebungen werden dabei erfahrungsgemäß häufig genannt. Das Beispiel der Farben ist natürlich nur dann einzubeziehen, wenn es behandelt wurde.

<sup>46</sup>Ein Ziel des auf S. 192f. skizzierten Ansatzes von BÜRGER et al. bestand darin, diese Unterscheidung überflüssig zu machen (vgl. auch [247], S. 163). Jedoch zeigten Untersuchungen von WITTMANN, dass dies auch in Kursen, in denen Vektoren arithmetisch eingeführt wurden, nicht gelang – die Schüler interpretierten Vektoren je nach dem Kontext ihrer Verwendung unterschiedlich ([276], S. 373f.).

<sup>47</sup>WITTMANN untersuchte auf Parametergleichungen von Geraden bezogene Schülerkonzepte und stellte fest, dass Schüler diese oft nicht als Gleichungen ansahen, die Mengen von Punkten in Abhängigkeit von Parametern beschreiben. Teilweise erkannten die Schüler weder die Bedeutung des Gleichheitszeichens in einer Parametergleichung noch die des Parameters, sondern betrachteten lediglich den Aufpunkt und den Richtungsvektor als „kennzeichnend“ für die beschriebene Gerade, vgl. [247], S. 140ff. und [276], S. 377ff. WITTMANN stellte heraus, dass diese eingeschränkte Betrachtungsweise genügen kann, um die im Unterricht behandelten Standardaufgaben zu bearbeiten ([276], S. 389f.).

Erkennen dieses Zusammenhangs setzt eine Sicht auf geometrische Objekte als Punktmengen voraus, geht aber darüber noch insofern hinaus, als die Abhängigkeit der Lage von Punkten im Raum von dem Parameter bzw. den Parametern erfasst wird.<sup>48</sup>

Als didaktische Ansätze, die Herausbildung auf den konkret-gegenständlichen Aspekt eingengter Konzepte von Parameterdarstellungen bei Schülern zu vermeiden sowie den Punktmengengedanken und den Schwerpunkt des funktionalen Zusammenhangs stärker einzubeziehen, bieten sich vor allem zwei Herangehensweisen an:

- (1) Die Schüler *konstruieren die zu einigen Parameterwerten gehörenden Punkte* bei einer Parameterdarstellung der Form  $P = P_0 + t \cdot \vec{a}$ ; und erkennen dabei, dass diese Punkte auf einer Geraden liegen. Davon ausgehend kann die Parameterbeschreibung von Geraden eingeführt werden;<sup>49</sup> auch die parameterabhängige Darstellung anderer Kurven ist so möglich. Des Weiteren bieten sich Umkehrüberlegungen an, bei denen zu einzelnen Punkten von Geraden bzw. Kurven ermittelt wird, welchem Wert des Parameters sie zugeordnet sind. Vergleiche verschiedener Parametrisierungen derselben Objekte erscheinen ebenfalls sinnvoll.
- (2) Es lässt sich die *dynamische Sicht auf Geraden und andere Kurven als Bahnkurven* hervorheben, wodurch die Schüler mit dem Parameter eine konkrete Bedeutung verbinden können.<sup>50</sup> Die Interpretation des Parameters als Zeit stellt dabei Bezüge zur Beschreibung von Bewegungen in der Physik her.

Es ist zu befürchten, dass die Berücksichtigung dieser beiden Herangehensweisen nur ansatzweise vermeidet, dass Schüler auf konkret-gegenständliche Aspekte beschränkte Vorstellungen herausbilden, wenn lediglich Geraden und Ebenen behandelt werden. Die Betrachtung linearer Objekte dürfte nicht ausreichen, um die Tragweite der Beschreibung geometrischer Objekte durch Parameterdarstellungen und die dabei auftretenden funktionalen Zusammenhänge und dynamischen Aspekte zu erfassen. Die *Behandlung von Kurven und evtl. auch Flächen* ist aus diesem Grunde nicht nur sinnvoll, um die

<sup>48</sup>Überlegungen von VOLLRATH zum funktionalen Denken (siehe [254]) konkretisierend, beschrieb WITTMANN in [276], S. 381f. drei Aspekte funktionalen Denkens bei der Arbeit mit Parameterdarstellungen von Geraden. Der *Zuordnungscharakter* lässt sich durch die Betrachtung von Geraden als Punktmengen und die Zuordnung von Punkten zu Parameterwerten berücksichtigen. Der Aspekt des *Änderungsverhaltens* impliziert eine dynamische Sichtweise, auf die noch eingegangen wird. Die *Sicht als Ganzes*, d. h. als durch eine Parametergleichung gegebenes Objekt kommt u. a. bei der Bestimmung von Schnittpunkten sowie bei der Betrachtung aus Geraden zusammengesetzter Objekte zum Tragen.

<sup>49</sup>Auch wenn Parameterdarstellungen von Geraden in der beschriebenen Weise eingeführt werden und den Schülern der „Punktmengengedanke“ dabei plausibel wird, ist es allerdings wahrscheinlich, dass dieses Konzept bei der weiteren Arbeit „verblasst“, wenn sich diese auf die Bearbeitung der genannten Standardaufgaben beschränkt. Es ist zu erwarten, dass sich hierbei eine Entwicklung vollzieht, die bereits hinsichtlich des Vektorkonzepts von Schülern beschrieben wurde – die Einführung von Gegenständen hat einen vergleichsweise geringen Stellenwert gegenüber den Kontexten der Arbeit damit im weiteren Verlauf des Unterrichts. Um die Punktmengenauffassung geometrischer Objekte bei den Schülern zu verankern, muss sie daher im folgenden Unterricht zur Anwendung kommen.

<sup>50</sup>MAASS schlug dazu in [170] vor, Bewegungsbahnen von Flugzeugen zu untersuchen.

in Abschnitt 1.4.5 diskutierte Formenarmut des Unterrichts in analytischer Geometrie zu überwinden, sondern bildet auch eine wichtige Bedingung dafür, dass Schüler die Idee des funktionalen Zusammenhangs im Zusammenhang mit Parameterdarstellungen tiefer gehend erfassen. Dafür sprechen vor allem folgende Gründe:

- Bei nichtlinearen Objekten liegen ganzheitliche Vorstellungen weniger nahe als bei Geraden; so existiert z. B. kein einzelner Vektor, der (analog zu einem Richtungsvektor einer Geraden) den Verlauf einer Kurve beschreibt.
- Durch die Betrachtung „neuer“ Objekte ergeben sich Anlässe, geometrische Objekte und algebraische Beschreibungen miteinander zu vergleichen und dabei Aspekte der Beschreibung von Objekten zu reflektieren. Die Gefahr, dass sich eingeschränkte Vorstellungen von Parameterdarstellungen verhärteten, die vor allem an der Abarbeitung von Kalkülen orientiert sind, wird dadurch verringert.

Die Einbeziehung von Elementen der Computergrafik erleichtert es, die auf S. 201 genannten Herangehensweisen (1) und (2) umzusetzen und schafft zudem günstige Voraussetzungen für die Behandlung durch Parameterdarstellungen beschriebener nichtlinearer Objekte. Die „Konstruktion“ von Geraden und Kurven sowie auch von Ebenen aus Punkten erfordert lediglich eine Software, mithilfe derer die Schüler entsprechende grafische Darstellungen anfertigen können.<sup>51</sup> Konkrete Vorschläge zur Berücksichtigung der Punktmengenauffassung bei der Behandlung der Parameterdarstellungen von Geraden enthält Abschnitt 4.2.2, auf Kurven wird in Abschnitt 4.3 eingegangen.

Für die *Herausbildung einer dynamischen Sicht auf Parameterdarstellungen* eignet sich m. E. in ausgezeichnete Weise die *Erstellung von Animationen* (Videos) in einer geeigneten Grafiksoftware oder einem CAS. Hierzu werden Positionen von Objekten oder auch die Position des Beobachters (bzw. der Kamera) in Abhängigkeit von einem Zeitparameter beschrieben (siehe Abschnitt 2.6). Da für die Erlangung eines Überblicks über den Ablauf von Animationen oft die Darstellung der verwendeten Bewegungsbahnen sinnvoll ist, können Vorgehensweisen zur Darstellung von Geraden bzw. Kurven als Punktmengen und dynamische Aspekte gut miteinander kombiniert werden. Ein wichtiges Argument für die Anfertigung von Animationen zur Erlangung einer dynamischen Sicht auf Parameterdarstellungen besteht darin, dass sich Schüler erfahrungsgemäß für die Generierung von Videos in überdurchschnittlich starkem Maße interessieren. Bei der Verwendung einer geeigneten Software sind dazu parameterabhängige Beschreibungen zwingend erforderlich. Angesichts der hier geschilderten Vorüberlegungen sollte die Möglichkeit, durch parameterabhängige Animationen Videos zu generieren, als Kriterium für die Auswahl der im Unterricht zu verwendenden Software Berücksichtigung finden. Unterrichtsvorschläge für die Einbeziehung von Animationen bei der Behandlung von Parameterdarstellungen enthält Abschnitt 4.3.

---

<sup>51</sup>Für Geraden in der Ebene oder ebene Kurven können grafische Darstellungen natürlich auch manuell angefertigt werden; die Übertragung auf den Raum dürfte durch Analogieüberlegungen gelingen. Soll hingegen von Anfang an im Dreidimensionalen gearbeitet werden, so ist die Nutzung des Computers erforderlich, da geeignete „materielle“ Modelle kaum zur Verfügung stehen.



## 3.4 Auswahl geeigneter Software für den Unterricht in analytischer Geometrie

### 3.4.1 Auswahlkriterien

In Abschnitt 2.7 wurde ein Überblick über Software zur Erstellung dreidimensionaler computergrafischer Darstellungen gegeben. Im Folgenden ist zu diskutieren, welche Softwarepakete hinsichtlich ihres Funktionsumfangs und ihrer Funktionsweise sowie der Bedienbarkeit für die Nutzung durch Schüler im Unterricht der analytischen Geometrie des Raumes geeignet sind. Dazu sind vor allem folgende Anforderungen zu stellen:

1. Die analytische Beschreibung geometrischer Objekte durch Koordinaten und Gleichungen ist unabdingbar; Programme, die lediglich Konstruktionen mit der Maus zulassen, sind für den Unterricht in analytischer Geometrie ungeeignet.
2. Die Möglichkeit, die derzeit im Unterricht behandelten Objekte (Pfeile als Repräsentanten von Vektoren, Geraden, Ebenen, Kugeln) darzustellen, muss gegeben sein. Wünschenswert erscheint es darüber hinaus, auch andere geometrische Körper sowie Flächen, die durch Gleichungen bzw. Parameterdarstellungen gegeben sind, darstellen sowie Schnittfiguren veranschaulichen zu können.
3. Die Qualität der Grafikausgabe durch die Software muss ausreichend sein, um eine räumliche Vorstellung von der Anordnung der dargestellten Objekte im Raum und ihrer gegenseitigen Lage zu erhalten. Dazu ist es auch notwendig, räumliche Szenen aus verschiedenen Richtungen betrachten zu können; zudem sollte die Möglichkeit gegeben sein, ein Koordinatensystem einzublenden.
4. Software, die im Mathematikunterricht verwendet wird, darf keinen zu hohen Einarbeitungsaufwand erfordern, da für das Erlernen des Umgangs mit einer Software im MU nur ein sehr begrenztes Maß an Zeit zur Verfügung gestellt werden kann.<sup>52</sup>
5. Es kommen nur Softwarepakete in Frage, deren Verwendung Informatikkenntnisse *nicht zwingend* voraussetzt.<sup>53</sup> Da es andererseits durchaus sinnvoll erscheint, einfache Prozeduren zu erstellen, wäre die *Möglichkeit* der Nutzung elementarer Programmierstrukturen wie Verzweigungen und Schleifen wünschenswert.

Neben diesen Grundvoraussetzungen erscheint es im Hinblick auf Ziele, Herangehensweisen und mögliche Gegenstandsbereiche der Einbeziehung von Elementen der Computergrafik in den Unterricht der analytischen Geometrie sinnvoll, folgende Merkmale für die Auswahl einer Software in Betracht zu ziehen:

---

<sup>52</sup>Die Schüler sollten nach einer geringen Einarbeitungszeit in der Lage sein, diejenigen Funktionen der Software zu nutzen, die sie für die im Kontext der behandelten mathematischen Inhalte zu lösenden Aufgaben benötigen. Unproblematisch ist es, wenn die verwendete Software zusätzliche Möglichkeiten bietet, deren Nutzung einen höheren Zeitaufwand erfordert, was z. B. bei CAS generell der Fall ist.

<sup>53</sup>Mit Informatikkenntnissen sind hierbei Kenntnisse hinsichtlich der Erstellung von Algorithmen (die über die Inhalte des Mathematikunterrichts hinausgehen) sowie Programmierkenntnisse gemeint. Da Informatik in keinem Bundesland (mit Ausnahme von Bayern) ein Pflichtfach ist, muss davon ausgegangen werden, dass viele Schüler über derartige Kenntnisse und Fähigkeiten nicht verfügen.

6. Wünschenswert erscheint es, dass die Schüler mithilfe der verwendeten Software Animationen durch die parameterabhängige Beschreibung von Positionen sowie Transformationen erstellen können (vgl. die Ausführungen auf S. 202).
7. Ästhetisch ansprechende Ergebnisse können eine Motivierung der Schüler bewirken, weshalb eine hohe Qualität der durch die Software berechneten Grafiken von Bedeutung ist. Diese lässt sich durch fotorealistische Bildberechnung erreichen, aber auch „synthetisch“ wirkende Grafiken können für Schüler reizvoll sein.
8. Der Überblick über dreidimensionale Szenen wird durch schnelle Rendering-Verfahren erleichtert, die Bilder in Echtzeit berechnen und somit die interaktive Navigation in Szenen ermöglichen.
9. Sollen mathematische Grundlagen der Bildberechnung im Unterricht thematisiert werden, so empfiehlt sich die Verwendung einer Software, die ein relativ einfach verständliches Rendering-Verfahren nutzt.

In Kapitel 2 wurde bereits deutlich, dass sich die Eigenschaften 7, 8 und 9 teilweise gegenseitig ausschließen.<sup>54</sup> Es sind also bei der Auswahl der zu nutzenden Software Kompromisse einzugehen, die im Folgenden diskutiert werden.

### 3.4.2 Eignung verschiedener Kategorien von Software für die Erstellung von 3D-Computergrafiken hinsichtlich ihres Einsatzes im Mathematikunterricht

Aufgrund von Kriterium 1 erscheint es nicht sinnvoll, die gebräuchlichen 3D-Grafikprogramme mit grafischen Benutzeroberflächen (vgl. Abschnitt 2.7.3), CAD-Systeme (2.7.4) und Software für den Unterricht in (synthetischer) Raumgeometrie (2.7.7)<sup>55</sup> in die engere Wahl für den Einsatz im Stoffgebiet Analytische Geometrie zu ziehen. Auch wenn einige dieser Programme Veränderungen von Koordinaten und einige 3D-Grafikprogramme sogar die Darstellung durch Parameterdarstellungen gegebener Flächen zulassen, so sind doch die hauptsächlichen Konstruktionsverfahren mausgestützt. Für die Nutzung im Unterricht der analytischen Geometrie ist jedoch m. E. eine Software erforderlich, welche für die Konstruktion geometrischer Objekte deren Beschreibung durch Koordinaten und Gleichungen voraussetzt. Diese Grundbedingung wird durch

- Computeralgebrasysteme (vgl. 2.7.5),
- spezielle Software für den Unterricht in analytischer Geometrie (2.7.6),

<sup>54</sup>Während z. B. das Raytracing-Verfahren eine hochwertige, fotorealistische Bildberechnung vornimmt und seine prinzipielle Funktionsweise recht leicht verständlich ist, ermöglicht es keine Echtzeit-Darstellung. Hingegen ist die erzielbare Qualität bei Verfahren, die Bilder in Echtzeit berechnen, begrenzt und ihre Funktionsweise aufgrund geschwindigkeitsbedingter Optimierungen komplizierter.

<sup>55</sup>Eine Ausnahme bildet diesbezüglich dynamische Raumgeometriesoftware, die Objekte auch durch analytische Beschreibungen generieren kann; derzeit (2007) trifft dies nur auf Archimedes Geo3D zu (siehe Abschnitt 2.7.7). Allerdings ist auch bei dieser Software die interaktive Erzeugung von Objekten durch Klicken und Ziehen mit der Maus die primäre Bedienungsweise. Erst bei Objekten, die sich auf diese Weise nicht generieren lassen, müssen dann analytische Beschreibungen genutzt werden.

- Programmiersprachen<sup>56</sup> und Szenenbeschreibungssprachen (SDL) sowie
- durch SDL gesteuerte Raytracing-Software (2.7.2)

erfüllt. Die Auswahl einer Software aus diesen Kategorien hängt vom Umfang sowie den Zielen der Einbeziehung von Elementen der Computergrafik in den Unterricht ab.

Sollen lediglich zu Standardthemen der vektoriellen Raumgeometrie Visualisierungen angefertigt werden, so lässt sich dies mit speziell für den Unterricht in analytischer Geometrie entwickelten Programmen wie *DreiDGeo* und *Vector3D* (siehe 2.7.6) mit geringem Aufwand erreichen. Der Umgang mit dieser Software ist sehr einfach zu erlernen, ihr Funktionsumfang jedoch stark begrenzt. Die Qualität der Grafikdarstellung entspricht dem hauptsächlichen Verwendungszweck, also der Veranschaulichung von Lagebeziehungen zwischen Geraden und Ebenen, die durch die Möglichkeit des Drehens und Skalierens der Ansicht erleichtert wird. Fragen der Wirkungsweise der Bildberechnung in der Computergrafik können anhand von *DreiDGeo* und ähnlichen Programmen nicht thematisiert werden. Auch zu Zwecken der Veranschaulichung klassischer Inhalte des Stoffgebietes Analytische Geometrie ist der Nutzung derartiger Software aus heutiger Sicht nicht mehr unbedingt der Vorzug zu geben<sup>57</sup> – da die Grafikfähigkeiten einiger CAS in den letzten Jahren stark verbessert wurden und CAS bereits häufig in anderen Themengebieten des Mathematikunterrichts zum Einsatz kommen, liegt es nahe, diese auch für Visualisierungen und Berechnungen in der analytischen Geometrie zu nutzen.<sup>58</sup>

Während Programme wie *DreiDGeo* oder *Archimedes Geo3D* für die durch analytische Beschreibungen darstellbaren Objekte Eingabemasken zur Verfügung stellen, erfordern es Szenenbeschreibungssprachen (SDL) und CAS,<sup>59</sup> grafisch darzustellende Objekte durch Textanweisungen zu beschreiben, entsprechende Beispiele enthält u. a. Abschnitt 2.7.2. Dies bedingt einen etwas höheren Aufwand für die Einarbeitung, da die Schüler Syntaxelemente der jeweiligen Sprache richtig verwenden müssen und z. B. fehlende oder

---

<sup>56</sup>Da allerdings, wie bereits mehrfach erwähnt wurde, die Programmierung von grundlegenden Prozeduren der Grafikdarstellung den Rahmen des Mathematikunterrichts weit überschreiten würde, erscheinen nur solche Programmiersprachen geeignet, die durch entsprechende Bibliotheken die Darstellung von Objekten durch einfache Befehle zulassen (siehe die Kriterien 4 und 5 auf S. 203). Ein Beispiel hierfür ist die Sprache Python in Verbindung mit dem Modul VPython, vgl. 2.6.5 und 2.7.2.

<sup>57</sup>Wie aus Abschnitt 2.7.6 hervorgeht, wurden *DreiDGeo* und einige ähnliche Programme bereits seit mehreren Jahren nicht mehr weiterentwickelt, was sicherlich u. a. darauf zurückzuführen ist, dass die Aufgaben, für die sie entwickelt wurden, inzwischen auch von CAS ausgeführt werden können.

<sup>58</sup>Eine weitere Alternative stellt seit kurzem die dynamische Raumgeometriesoftware *Archimedes Geo3D* dar. Insbesondere falls Schüler diese Software bereits kennen, so können Sie mit Ihrer Hilfe leicht z. B. Lagebeziehungen von Geraden und Ebenen veranschaulichen. Wie in Abschnitt 2.7.7 erwähnt wurde, lassen sich in *Archimedes Geo3D* darüber hinaus auch Kurven und Flächen durch analytische Beschreibungen darstellen. Das Vorgehen hierbei ist – insbesondere wenn Variationen vorgenommen werden sollen – allerdings nicht einfacher als in CAS oder Szenenbeschreibungssprachen.

<sup>59</sup>Diese Aussage trifft auf die leistungsfähigeren CAS Maple, Mathematica und MuPAD zu, wohingegen *Derive* und *Mathcad* Menübefehle zur Erstellung von Grafiken enthalten. Allerdings sind gerade diese beiden Systeme, vor allem *Derive*, hinsichtlich ihrer grafischen Möglichkeiten recht begrenzt (siehe Abschnitt 2.7.5) und werden deshalb in der vorliegenden Arbeit nicht weiter betrachtet.

falsche Klammern zu Fehlermeldungen führen. Allerdings hat sich in Unterrichtsversuchen gezeigt, dass die Schüler bereits nach wenigen Minuten der Arbeit mit Programmen wie MuPAD und POV-Ray in der Lage waren, Anweisungen richtig einzugeben. Eine wichtige Voraussetzung für die schnelle Einarbeitung in Skript- bzw. Beschreibungssprachen und CAS besteht darin, den Schülern Anleitungen zur Verfügung zu stellen, welche die jeweils benötigten Sprachelemente in kurzer und prägnanter Weise erläutern.<sup>60</sup> Stehen derartige Anleitungen zur Verfügung, so ist der Einarbeitungsaufwand für die Erstellung grafischer Darstellungen in Skriptsprachen, SDL und CAS nur unwesentlich höher als bei der Verwendung von Eingabemasken. Die auf S. 203 genannten Kriterien 4 und 5 sind somit auch bei der Verwendung von Software, die das Erlernen von Syntaxelementen erfordert, erfüllbar.<sup>61</sup> Dennoch sollte auf eine möglichst einfache Syntax der verwendeten Skript- bzw. Beschreibungssprache geachtet werden.

Wie anhand des in Abschnitt 2.7.2 mithilfe verschiedener Beschreibungssprachen dargestellten Beispiels deutlich wurde, ermöglichen das Modul *VPython* für die Programmiersprache Python, die SDL der Raytracing-Software *POV-Ray* und die in dem CAS *MuPAD* vorhandenen Befehle eine recht einfache Beschreibung von geometrischen Objekten durch Koordinaten, weshalb diese Softwarepakete m. E. für die Verwendung im Unterricht der analytischen Geometrie in die engere Wahl zu ziehen sind.<sup>62</sup>

Gegenüber POV-Ray und MuPAD ist der Funktionsumfang von *VPython* sowohl hinsichtlich der darstellbaren geometrischen Objekte<sup>63</sup> als auch in Bezug auf die grafischen Eigenschaften begrenzt; auch die Qualität der grafischen Ausgabe reicht nicht an die anderer Softwarepakete heran. Da *VPython* jedoch lediglich ein Ergänzungsmodul für die leistungsfähige Programmiersprache Python ist, böten sich umfangreiche Möglichkeiten, wenn die Schüler über Programmierfähigkeiten mithilfe von Python verfügten.

<sup>60</sup>Auf entsprechende Anleitungen für die Arbeit mit POV-Ray wird in den Abschnitten 3.5 und 4.2 noch eingegangen, siehe auch Anhang A.

<sup>61</sup>Zudem bieten derartige Programme den Vorteil, dass die Beschreibungen aller dargestellten Objekte (einschließlich ihrer Koordinaten bzw. Gleichungen) gleichzeitig überblickt werden können. Demgegenüber ist die Fehlersuche in Programmen, die mit Eingabemasken bzw. Dialogfeldern arbeiten, aufwändiger, da diese für jedes zu überprüfende Objekt geöffnet werden müssen, um dessen Koordinaten und Gleichungen zu betrachten. Insbesondere für Lehrer ist es oft von Vorteil, „auf einen Blick“ die Eingaben von Schülern erfassen zu können, falls Probleme auftreten und Schüler Hilfe benötigen.

<sup>62</sup>Die hierarchische Modellierungssprache VRML, der die Struktur von Szenengraphen (in denen u. a. geometrische Objekte sowie ihre Eigenschaften Knoten bilden) zugrunde liegt, verwendet eine vergleichsweise komplizierte Syntax (vgl. das Beispiel auf S. 163). Da die Verwendung von VRML gegenüber beispielsweise *VPython* oder *MuPAD* für die Nutzung im Stoffgebiet Analytische Geometrie keine erkennbaren Vorteile aufweist, wird diese Sprache nicht in die engere Wahl gezogen. Ebenso wenig erscheint es sinnvoll, verschiedene durch SDL gesteuerte Raytracing-Systeme (siehe S. 164f.) zu betrachten, die gegenüber POV-Ray lediglich hinsichtlich ihrer Netzwerkfähigkeit sowie weniger Spezialeffekte Vorteile aufweisen, deren Bedienung jedoch aufwändiger ist.

<sup>63</sup>*VPython* ermöglicht die Darstellung von Streckenzügen, Pfeilen, Zylindern, Kegeln, Pyramiden, Quadern, Kugeln, Tori, Ellipsoiden sowie ebenen Kurven (in der  $x$ - $y$ -Ebene). Es ist jedoch nicht möglich, durch Gleichungen oder Parameterdarstellungen gegebene Flächen darzustellen. Dazu wäre es erforderlich, entsprechende Erweiterungsmodule in Python zu programmieren oder bereits vorhandene Python-Zusatzmodule zu nutzen, die allerdings nicht in Kombination mit *VPython* einsetzbar sind.

Allerdings würde das Erlernen auch dieser verhältnismäßig einfachen Programmiersprache über den Rahmen des Mathematikunterrichts weit hinausgehen. Die Nutzung von Python und VPython wäre also vor allem dann sinnvoll, wenn dies fächerübergreifend zwischen Mathematik- und Informatikunterricht erfolgte. Im Rahmen eines solchen fächerübergreifenden Unterrichts könnten die Schüler unter Anwendung von Inhalten der analytischen Geometrie, der Informatik und evtl. zusätzlich auch der Physik mithilfe von Python und VPython u. a. komplexere Simulationen oder auch einfache Spiele mit 3D-Grafikausgabe erstellen. Wie allerdings bereits erwähnt wurde, dürfte eine enge Kooperation zwischen Mathematik- und Informatikunterricht nur in Ausnahmefällen möglich sein, da meist nur eine Minderheit der Schüler Informatikkurse belegt.

### 3.4.3 Vor- und Nachteile von CAS und Raytracing-Software

Für die Einbeziehung von Elementen der 3D-Computergrafik in den Unterricht der analytischen Geometrie ohne Kooperation mit dem Fach Informatik bieten sich nach den geschilderten Überlegungen vor allem CAS mit leistungsfähigen Grafikfunktionen<sup>64</sup> sowie die Raytracing-Software POV-Ray an. Beide Systeme werden den auf S. 203 genannten Anforderungen 1 und 2 gerecht, wie viele Beispiele in Kapitel 2, die mithilfe von POV-Ray und des CAS MuPAD angefertigt wurden, zeigen. Auch Bedingung 3 wird von beiden Systemen erfüllt, wobei allerdings Unterschiede hinsichtlich der interaktiven Navigationsmöglichkeiten in Grafiken bestehen, auf die noch einzugehen ist. Ein Koordinatensystem zeigt MuPAD in Grafiken standardmäßig an, bei Verwendung von POV-Ray müssen dafür angefertigte Zusatzbibliotheken verwendet werden.

Eine schnelle Einarbeitung in die Nutzung der grafischen Möglichkeiten (Bedingung 4) erfordert, wie bereits erwähnt wurde, geeignete Anleitungen für die Schüler; für die Arbeit mit POV-Ray sollten zusätzlich Vorlagedateien bereitgestellt werden (siehe Abschnitt 3.5 und Anhang A). Hinsichtlich Bedingung 5 haben etliche Beispiele in Kapitel 2 bereits gezeigt, dass sowohl die Nutzung von POV-Ray als auch die von MuPAD ohne spezielle Vorkenntnisse aus der Informatik möglich ist. Beide Programme bieten aber auch Möglichkeiten des Programmierens; fächerübergreifende Bezüge zum Informatikunterricht sind zwar nicht zwingend erforderlich, lassen sich aber gut herstellen.<sup>65</sup> Auch die Erstellung von Animationen (siehe Kriterium 6 auf S. 204) ist sowohl mithilfe von POV-Ray als auch in MuPAD möglich, wenngleich sich die technischen Vorgehensweisen dabei zwischen beiden Programmen erheblich unterscheiden.

---

<sup>64</sup>Im Folgenden wird das CAS MuPAD betrachtet, das seit der Version 3 über sehr leistungsfähige Grafikfunktionen verfügt und u. a. leicht nutzbare Befehle für die Darstellung geometrischer Grundkörper enthält. Allerdings könnten für viele der unterbreiteten Vorschläge, welche die Darstellung von Geraden, Ebenen, Kurven und Flächen beinhalten, auch andere CAS wie Maple oder Mathematica genutzt werden. Wie bereits erwähnt wurde, wären jedoch bei der Nutzung von Derive Einschränkungen hinzunehmen (siehe Abschnitt 2.7.5).

<sup>65</sup>Insbesondere POV-Ray wird auch in einer Reihe von Informatikkursen genutzt, wie anhand entsprechender Internetseiten deutlich wird. Eine entsprechende Linksammlung enthält die Seite von F. LOHMÜLLER: [www.f-lohmueller.de/links/index\\_sd.htm#school](http://www.f-lohmueller.de/links/index_sd.htm#school).

Grundsätzliche Unterschiede bestehen zwischen POV-Ray und den Grafikfunktionen von MuPAD und anderen CAS hinsichtlich des verwendeten Verfahrens der Bildberechnung. Während POV-Ray das vergleichsweise langsame Raytracing-Verfahren nutzt, welches fotorealistische Bilder generieren kann, berechnet MuPAD grafische Darstellungen durch Scanline-Rendering unter Verwendung der OpenGL-Bibliotheken (vgl. die Abschnitte 2.5.4, 2.5.5 und 2.7). Daraus ergeben sich wesentliche Unterschiede zwischen diesen Programmen hinsichtlich der Bedingungen 7, 8 und 9 (S. 204):

- Während es MuPAD durch Echtzeit-Rendering ermöglicht, geometrische Objekte und Szenen interaktiv (durch Ziehen mit der Maus) aus verschiedenen Richtungen zu betrachten sowie zu vergrößern und zu verkleinern, erfordert die Darstellung verschiedener Ansichten mithilfe von POV-Ray jeweils die Veränderung von Kamerakoordinaten bzw. des Öffnungswinkels und eine Neuberechnung des Bildes. Auch wenn dies in vielen Fällen nur wenige Sekunden dauert, kann sich dadurch z. B. die für die Positionierung von Objekten benötigte Zeit signifikant verlängern. Das Fehlen einer interaktiven Darstellung ist somit ein Nachteil von POV-Ray, der sich auch dann besonders auswirkt, wenn möglichst genau Winkel oder Positionen von Schnittobjekten abgeschätzt werden sollen. Dazu sind meist mehrere Bildberechnungen erforderlich, während in interaktiven MuPAD-Grafiken recht schnell eine günstige Perspektive gefunden werden kann.<sup>66</sup>
- Sowohl MuPAD als auch POV-Ray können geometrische Objekte in hoher Qualität grafisch darstellen. MuPAD ermöglicht auch die stilisierte Darstellung und Animation „realer“ Objekte und Szenen;<sup>67</sup> allerdings ist es mit dieser Software (im Gegensatz zu POV-Ray) prinzipbedingt nicht möglich, „fotorealistische“ Grafiken anzufertigen, wie sie Schülern aus vielen Bereichen ihrer Lebensumgebung vertraut sind. Fotorealistische Computergrafik ist ein Phänomen, welches Schüler nicht a priori mit mathematischen Grundlagen in Verbindung bringen. Die Nutzung einer Software, die nicht vorrangig der Visualisierung mathematischer Zusammenhänge dient und die Erstellung von Grafiken, deren Aussehen nicht dem entspricht, was Schüler unter „mathematischen Illustrationen“ erwarten, kann sie überraschen, ihre Neugier wecken und insbesondere auch auf Schüler, die sich nur mäßig für Mathematik interessieren, motivierend wirken.<sup>68</sup> Daraus kann jedoch

<sup>66</sup>In einigen Situationen kann fehlende Interaktivität aus didaktischer Sicht vorteilhaft sein: Der höhere Aufwand für die Betrachtung von Szenen aus verschiedenen Richtungen erfordert genauere Überlegungen bei der Wahl sinnvoller Koordinaten und dürfte somit den Anteil von den Schülern zu vollziehender „Kopfgeometrie“ erhöhen. Wie in Abschnitt 1.4.4 ausgeführt wurde, erfordert die Erlangung einer „koordinatenbezogenen Raumvorstellung“ nicht nur visuelles, sondern vor allem aktives gedankliches Operieren mit Koordinaten und zugehörigen Positionen. Allerdings sind auch bei der Verwendung von CAS mit interaktiver Grafikdarstellung aufgrund der notwendigen Beschreibung geometrischer Objekte durch Koordinaten bzw. Gleichungen Vorüberlegungen erforderlich, so dass fehlende Interaktivität auch unter diesem Aspekt nur sehr bedingt als Vorteil anzusehen ist.

<sup>67</sup>Entsprechende Beispiele, die teilweise von Schülern angefertigt wurden, stehen auf der Internetseite <http://www.sciface.com/education/material.php> zur Verfügung.

<sup>68</sup>Zu erwarten wäre dabei evtl. die Frage von den Schülern, warum keine Grafikprogramme mit

für den Mathematikunterricht nur dann ein Nutzen resultieren, wenn es gelingt, so hervorgerufene Motivation in anspruchsvolles mathematisches Arbeiten umzusetzen. Dafür Wege aufzuzeigen, ist ein wichtiges Ziel dieser Arbeit.

- Die Erstellung von Animationen ist in MuPAD unkomplizierter und erfordert weniger Zeit als bei Nutzung von POV-Ray, das es allerdings ermöglicht, fotorealistische Animationen zu generieren, welche Schüler an „echte“ Videos erinnern.
- Die Funktionsweise des von der Software genutzten Rendering-Verfahrens ist nur dann von Bedeutung, falls mathematische Grundlagen der Bildberechnung in der Computergrafik im Unterricht thematisiert werden sollen, der auf S. 189 genannte Aspekt (G) also in stärkerem Maße zum Tragen kommt. In dieser Hinsicht bietet POV-Ray aufgrund des verwendeten Raytracing-Verfahrens erhebliche Vorteile. Da sich Raytracing an den elementaren Gesetzen der geometrischen Optik orientiert (siehe Abschnitt 2.5.5), ist seine Funktionsweise im Vergleich zu der anderer Verfahren recht leicht verständlich,<sup>69</sup> zudem beruhen die wesentlichen Schritte der Bildberechnung auf Inhalten der analytischen Geometrie, die nach den meisten heutigen Rahmenplänen Inhalte des Mathematikunterrichts sind. Aus diesen Gründen erscheint das Raytracing-Verfahren hinsichtlich einer Behandlung ausgewählter mathematischer Grundlagen der 3D-Computergrafik als Anwendungen von Gegenständen der analytischen Geometrie mit Schülern gut geeignet. Günstig ist dafür auch, dass geometrische Objekte (z. B. Kugeln, Zylinder und sogar Flächen, die durch implizite Gleichungen oder Parameterdarstellungen gegeben sind) von POV-Ray direkt verarbeitet werden, während CAS diese vor der grafischen Darstellung in Polygonnetze umwandeln (vgl. 2.3.3, 2.3.5 und 2.3.9). Das Raytracing-Verfahren benötigt für die Bildberechnung zudem deutlich weniger Verfahrensschritte als Grafik-Pipelines, die bei Scanline-Verfahren zum Einsatz kommen (siehe S. 84) und kann deshalb in seiner Gesamt-Wirkungsweise wesentlich leichter im Unterricht thematisiert und für die Schüler verständlich werden.

Zusammenfassend ist festzustellen, dass sowohl POV-Ray als auch ein CAS mit leistungsfähigen Grafikfunktionen wie MuPAD als Hilfsmittel für Visualisierungen und die Unterstützung experimentellen Arbeitens im Unterricht der analytischen Geometrie genutzt werden können, also dem auf S. 189 genannten Aspekt (W) der Einbeziehung von Elementen der 3D-Computergrafik in den Unterricht gerecht werden. Dabei weist MuPAD einige Vorteile auf, wie z. B. die Möglichkeit der interaktiven Betrachtung

---

Mausbedienung (vgl. 2.7.3) genutzt werden, bei denen Koordinatenbeschreibungen nicht nötig sind. Allerdings wurde diese Frage in Unterrichtsversuchen nie gestellt, obwohl zu Vergleichszwecken die Demonstration einer entsprechenden Software erfolgte. Dies hängt sicherlich damit zusammen, dass es Schüler als selbstverständlich ansehen, im Mathematikunterricht mit mathematischen Beschreibungen zu arbeiten. Zudem faszinierte es viele Schüler besonders, wie aus derartigen Beschreibungen „nicht nach Mathematik aussehende“ Objekte und Szenen hervorgingen. Auf diesbezügliche Erfahrungen und Schüleräußerungen wird in Kapitel 5 noch näher eingegangen.

<sup>69</sup>Raytracing ist zudem der natürlichste Zugang zu einer Vorstellung von der Beleuchtung computergenerierter Szenen, da die Verfolgung von Lichtstrahlen die Lichtausbreitung in der Realität simuliert und dem menschlichen Sehen entspricht.

dreidimensionaler Objekte und Szenen.<sup>70</sup> Demgegenüber bietet sich die Verwendung von POV-Ray vor allem dann an, wenn im Sinne des Aspektes (G) auch mathematische Grundlagen der Bildberechnung in der 3D-Computergrafik im Unterricht thematisiert werden sollen. Auch die motivierende Wirkung, welche die Anfertigung fotorealistischer Grafiken auf Schüler ausübt, spricht für die Nutzung von POV-Ray.<sup>71</sup>

Da mit der vorliegenden Arbeit das Ziel verfolgt wird, in möglichst umfassender Weise Potenzen der Einbeziehung von Elementen der 3D-Computergrafik in den Unterricht der analytischen Geometrie zu untersuchen und dieses Ziel die Berücksichtigung beider auf S. 189 aufgeführter Aspekte (W) und (G) erfordert, liegt den in Kapitel 4 unterbreiteten Vorschlägen hauptsächlich die Nutzung von POV-Ray zugrunde; auch bei den in Kapitel 5 beschriebenen Unterrichtsversuchen kam diese Software zum Einsatz. Für einen Teil der unterbreiteten Vorschläge könnte jedoch anstelle von POV-Ray auch ein CAS zum Einsatz kommen, was an entsprechenden Stellen erwähnt und teilweise anhand von Beispielen verdeutlicht wird.<sup>72</sup>

Hinsichtlich der Auswahl einer geeigneten Software können sich mittel- bis langfristig die Voraussetzungen ändern, so dass einige der in diesem Abschnitt angestellten Überlegungen möglicherweise in ca. 5 Jahren nicht mehr zutreffen.<sup>73</sup> Ein Anspruch an die in Kapitel 4 dargelegten Unterrichtsvorschläge und -konzepte muss also darin bestehen, in nicht zu starkem Maße auf die verwendete Software bezogen zu sein, so dass die grundlegenden Überlegungen auch auf künftig evtl. verfügbare Produkte anwendbar sind. Gleichzeitig erfordern aber konkrete, in der Praxis umsetzbare Vorschläge auch Überlegungen zur Nutzung der verwendeten Software durch die Schüler, dabei sind Materialien und Anleitungen zu entwickeln, die nicht unabhängig von der Software sein können, die im Unterricht zum Einsatz kommt.

---

<sup>70</sup>Ein weiterer Vorteil der Nutzung eines CAS gegenüber der eines reinen Grafikprogramms im Unterricht besteht darin, dass mithilfe von CAS auch Berechnungen durchgeführt, z. B. lineare Gleichungssysteme gelöst werden können. Zudem lassen sich CAS in praktisch allen Themengebieten des Mathematikunterrichts der SII einsetzen.

<sup>71</sup>Als zusätzliches Argument für die Nutzung von POV-Ray könnte die Tatsache angeführt werden, dass es sich um Freeware handelt, die kostenlos in der Schule und von den Schülern zu Hause genutzt werden kann. Allerdings ist eine Argumentation anhand der Preise im Zusammenhang mit konzeptionellen, die Weiterentwicklung des Unterrichts betreffenden Überlegungen m. E. nicht sinnvoll. Des Weiteren verfügen viele Schulen bereits über CAS; außerdem bietet die Firma SciFace das CAS MuPAD für Schulen zu recht günstigen Konditionen an; dabei existieren auch Lizenzmodelle, welche die häusliche Benutzung durch Schüler und Lehrer ermöglichen.

<sup>72</sup>Schließlich könnte für einfache Visualisierungen der nach den gegenwärtigen Rahmenplänen hauptsächlich behandelten Inhalte des Stoffgebietes Analytische Geometrie auch eine Software wie DreiDGeo oder Vectory genutzt werden. Konzeptionell ergeben sich dabei keine erheblichen Unterschiede zu einer Veranschaulichung von Vektoren, Geraden und Ebenen mithilfe von POV-Ray oder eines CAS.

<sup>73</sup>Wie bereits in Abschnitt 2.6.5 (S. 158) erwähnt wurde, existieren Forschungsprojekte, die sich mit der Entwicklung von Echtzeit-Raytracing-Systemen beschäftigen (siehe auch [167]). Mit der Verfügbarkeit entsprechender Produkte würde sich der Widerspruch zwischen fotorealistischer Bildqualität und interaktiver Betrachtung von 3D-Szenen auflösen. Allerdings ist m. E. nicht damit zu rechnen, dass derartige Software (die dann auch geeignete leistungsfähige Hardware erfordert), bis zum Jahre 2010 in breitem Umfang verfügbar sein wird.



### 3.5 Vorlagen und Anleitungen für die Nutzung von POV-Ray im Unterricht

Naturgemäß kann im Mathematikunterricht nicht viel Zeit für das Erlernen einer Software durch die Schüler aufgewendet werden. Vielmehr ist es notwendig, Voraussetzungen dafür zu schaffen, dass die Schüler die verwendete Software sehr schnell nutzen können, um Aufgaben zu bearbeiten, die mit den Unterrichtszielen im Zusammenhang stehen. Wie bereits erwähnt wurde, ist es bei der Verwendung von Skript- und Beschreibungssprachen dazu notwendig, kurze Anleitungen mit den benötigten Anweisungen zur Verfügung zu stellen.<sup>74</sup> Die Einarbeitung in die Software sollte auf deren Grundlage an Beispielen erfolgen, die ohnehin als Unterrichtsgegenstände vorgesehen sind.

Kurze Anleitungen für den Einstieg in die Nutzung der Software POV-Ray, für die Darstellung von Vektoren (als Pfeile), Geraden und Ebenen mithilfe dieser Software, geometrische Transformationen, die Darstellung von Booleschen Operationen, Prismen, Rotationskörpern und Flächen, die durch implizite Gleichungen oder Parameterdarstellungen gegeben sind, enthält Anhang A (S. 329ff.).<sup>75</sup> Außerdem ist eine Anleitung für die Erstellung von Animationen mithilfe von POV-Ray enthalten, die allerdings stärker technisch orientiert ist und den Bezug zum Unterrichtsstoff (Parameterdarstellungen) nur am Rande anspricht.<sup>76</sup> Die Anleitungen können unabhängig voneinander verwendet werden – mit der einzigen Ausnahme, dass die Schüler z. B. anhand der „Kurzanleitung für die Arbeit mit POV-Ray“ (siehe S. 330f.) mit der grundlegenden Bedienung des Programms vertraut geworden sein sollten, bevor sie die anderen Anleitungen nutzen.

---

<sup>74</sup>Die zu CAS sowie zu Grafikprogrammen wie POV-Ray gehörenden Hilfesysteme sind entsprechend dem Funktionsumfang und der Leistungsfähigkeit dieser Softwarepakete sehr umfangreich. Dies führt dazu, dass die ersten Schritte unter Verwendung der mitgelieferten Handbücher und Hilfedateien oft recht zeitaufwändig sind. Die in diesen Materialien enthaltenen Einführungslehrgänge (Tutorials) entsprechen zudem oft nicht den Inhalten, die im Mathematikunterricht behandelt werden sollen. Keinesfalls soll damit zum Ausdruck kommen, dass die Schüler nicht die Dokumentationen der Software sowie zusätzliche Informationen z. B. aus dem Internet nutzen sollten – es ist sogar anstrengenswert, dass sie sich auf diese Weise Informationen beschaffen, um ihre Vorstellungen zu realisieren. Für den Einstieg erscheint jedoch sowohl aus Zeitgründen als auch hinsichtlich der Fokussierung auf die Unterrichtsgegenstände die Erstellung danach ausgerichteter Anleitungen notwendig.

<sup>75</sup>Diese Anleitungen wurden für den Einsatz im Unterricht bzw. in Seminaren erstellt; im Zusammenhang mit konkreten Unterrichtsvorschlägen wird in Kapitel 4 noch darauf Bezug genommen. Sie beschränken sich auf die unbedingt notwendigen Befehle und decken nur einen sehr geringen Teil der Funktionalität von POV-Ray ab, da das Hauptaugenmerk bei der Erarbeitung der Anleitungen auf einem schnellen Einstieg in die Nutzung der Software für die Unterrichtsthemen lag. Weiter gehende Informationen für die Erstellung von Grafiken mithilfe von POV-Ray beschafften sich viele Schüler und Studierende aus der Hilfe sowie vor allem von der Internetseite von F. LOHMÜLLER: [www.f-lohmueller.de/pov\\_tut/pov\\_\\_ger.htm](http://www.f-lohmueller.de/pov_tut/pov__ger.htm), die ausführliche Anleitungen zur Nutzung sehr vieler Möglichkeiten von POV-Ray enthält, welche aber zum größten Teil weit über die Inhalte hinausgehen, die im Mathematikunterricht allein aus Zeitgründen behandelt werden können.

<sup>76</sup>Diese Anleitung entstand aus dem Wunsch vieler Schüler heraus, Videos zu erstellen. Da für die diesbezüglichen technischen Aspekte im Unterricht nicht genügend Zeit vorhanden war, konnten interessierte Schüler mithilfe dieser Anleitung in ihrer Freizeit an Videos arbeiten.

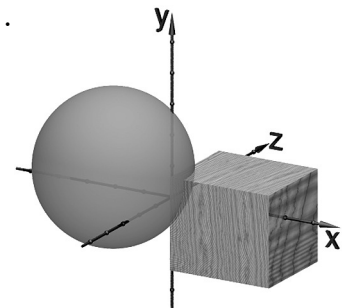
Wie bereits anhand des einführenden Beispiels auf S. 80 deutlich wurde, erfordert die Erstellung dreidimensionaler Szenen in POV-Ray neben der Beschreibung geometrischer Objekte zwingend die Festlegung einer *Kamera*, die Definition von *Lichtquellen* sowie die Zuweisung von *Texturen* zu geometrischen Objekten,<sup>77</sup> bevor Bilder berechnet werden können.<sup>78</sup> Die Gleichzeitigkeit dieser Anforderungen erschwert den Einstieg in die Software und führt zu längeren Einarbeitungszeiten. Um die Bedienung der Software anfangs zu erleichtern, erschien es deshalb sinnvoll, Vorlagen zu erstellen, in denen Kameras und Lichtquellen bereits vorbereitet sowie Texturen beschrieben sind, welche die Schüler Objekten anhand einfacher Namen zuordnen können. Da POV-Ray keine Möglichkeit bereitstellt, den berechneten Grafiken ein die Orientierung erleichterndes Koordinatenkreuz hinzuzufügen, kann aus diesen Vorlagen heraus (durch Aufruf eines Makros) zudem sehr leicht ein solches in Szenen eingefügt werden kann.

Für die Erstellung erster Szenen in POV-Ray können die Schüler eine Vorlagedatei mit folgendem Inhalt öffnen und durch Beschreibungen geometrischer Objekte ergänzen.<sup>79</sup>

```
// Vorlage fuer Szenen (vorlage.pov)
// -----
#version 3.5;           // benötigte POV-Ray-Version
#declare intervall = 3; // von der Kamera erfasster Bildausschnitt
#declare winkel = - 60; // Winkel der Kamera zur x-Achse (in Grad)
#declare hoehe = 4;    // Höhe (relativ) der Kamera über der x-z-Ebene
#include "vorlage.inc"  // vordefinierte Objekte und Materialien
background {White}     // Farbe des Hintergrundes
// -----
// Beginnen Sie, Objekte in der Szene zu beschreiben.
```

Es genügt, in die Vorlage folgende Anweisungen einzugeben, um die rechts abgebildete Grafik zu erzeugen:

```
ks
sphere { <-1.5,1,-1> 2.5 texture {blau_transp} }
box { <0.5,-2,-1> <4,1,2> texture {holz} }
```



Diese und weitere einfache Objekte sowie die Namen der Texturen sind in der Kurzanleitung auf S. 330f. erläutert.

Bei der Verwendung der Vorlage können drei Werte verändert werden, die den dargestellten Ausschnitt des Raumes sowie die Sicht auf die Szene beeinflussen:

<sup>77</sup>Erfolgt keine Zuweisung von Texturen, so rendert POV-Ray Szenen zwar ebenfalls, allerdings werden die Objekte dann gleichmäßig schwarz dargestellt. Da der Hintergrund ebenfalls schwarz erscheint, wenn keine anderen Festlegungen getroffen werden, ist auf dem Bild nichts zu erkennen.

<sup>78</sup>Zwar bietet POV-Ray über sein Insert-Menü bereits selbst die Möglichkeit, Befehle z. B. für die Festlegung von Kameras und Lichtquellen einzufügen; allerdings müssen dann immer noch die entsprechenden Koordinaten und Parameter festgelegt werden.

<sup>79</sup>Die Vorlagedatei ruft die Include-Datei `vorlage.inc` auf, in der die Beschreibungen von Kamera, Lichtquellen und Texturen sowie ein Makro für die Erstellung eines Koordinatenkreuzes enthalten sind. Die Include-Datei muss sich entweder in demselben Ordner befinden wie die Vorlage- bzw. die daraus erstellten Arbeitsdateien oder im Suchpfad von POV-Ray liegen (siehe S. 329). Am einfachsten ist es i. Allg., Vorlage-, Arbeits- und Include-Dateien in denselben Ordner zu kopieren.

**intervall** gibt an, wie groß der abgebildete Ausschnitt des Raumes ist. Die Kamera ist auf den Koordinatenursprung gerichtet und erfasst etwa alle Punkte des Raumes, deren sämtliche Koordinaten zwischen  $-\text{intervall}$  und  $\text{intervall}$  liegen.

**winkel** beschreibt den orientierten Winkel zwischen der  $x$ -Achse und dem auf die  $x$ - $z$ -Ebene projizierten Lot von der Kamera auf die  $y$ -Achse.

**hoehe** gibt die (relative, von **intervall** abhängige) Höhe der Kamera über der  $x$ - $z$ -Ebene an.

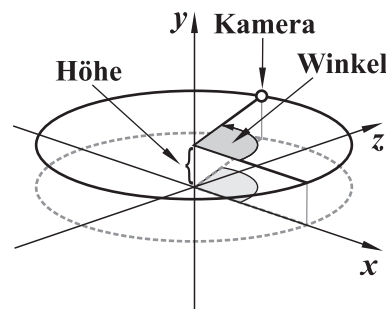


Abbildung 3.2: Kameraeinstellungen in der Vorlage

Die Werte dieser drei Variablen werden an die Include-Datei `vorlage.inc` übergeben, welche damit die Kamerakoordinaten

```
(60·intervall·cos(winkel); 5·hoehe·intervall; 60·intervall·sin(winkel))
```

festlegt.<sup>80</sup> Auf weitere Möglichkeiten, die Abbildung eines Raumausschnittes zu beeinflussen, wurde verzichtet, da zusätzliche Variable die Vorlage unübersichtlicher gemacht hätten. So ist die in der Vorlage definierte Kamera stets auf den Koordinatenursprung gerichtet. Jedoch ist es möglich, z. B. mittels

```
camera { location <x,y,z> angle  $\theta$  look_at < $x_L, y_L, z_L$ > }
```

eine Kamera selbst festzulegen; die oben beschriebene Kamera wird dann nicht verwendet. Nachdem Schüler einige Erfahrungen im Umgang mit POV-Ray gewonnen haben, können sie auch zusätzliche Lichtquellen festlegen, Texturen selbst definieren und schließlich auf die Verwendung der Vorlage ganz verzichten.

Ein mögliches Problem bei der Arbeit mit POV-Ray könnte darin bestehen, dass die Ausrichtung des (linkshändigen) Koordinatensystems von POV-Ray für Schüler ungewohnt ist. Prinzipiell bestünde die Möglichkeit, auf alle Objekte eine Transformationsmatrix anzuwenden (siehe Abschnitt 2.3.1, S. 85). Da sich allerdings in Unterrichtsversuchen zeigte, dass sich die Schüler sehr schnell an das Koordinatensystem von POV-Ray gewöhnten, erscheint es m. E. sinnvoller, auf eine solche Transformation zu verzichten, zumal die verwendete Matrix für Schüler am Anfang des Stoffgebietes Analytische Geometrie eine „black box“ darstellen würde.<sup>81</sup>

Bei mehrfachen Erprobungen der Datei `vorlage.pov` (im Zusammenhang mit der „Kurzanleitung für die Arbeit mit POV-Ray“, siehe S. 330f.) in Unterrichtsversuchen stellte sich heraus, dass die Schüler jeweils nach wenigen Minuten POV-Ray bedienen und erste Grafiken erstellen konnten. Weitere Vorlagen und Include-Dateien wurden für die Darstellung „klassischer“ Inhalte des Unterrichts in analytischer Geometrie (Punkte, Vektoren, Geraden und Ebenen) erstellt (siehe Abschnitt 4.2 und Anhang A).

<sup>80</sup>Außerdem ist die Größe des mittels **ks** anzuzeigenden Koordinatenkreuzes von der Variablen **intervall** abhängig, so dass es für beliebige Werte von **intervall** vollständig im Bild enthalten ist. Die konkrete Realisierung der Definitionen und Makros ist in Anhang A auf S. 341f. dokumentiert.

<sup>81</sup>Soll POV-Ray allerdings für die Veranschaulichung von Aufgaben aus Schulbüchern genutzt werden, so kann im Interesse der Übereinstimmung mit den dortigen Abbildungen eine Transformation in das Koordinatensystem des verwendeten Lehrbuches sinnvoll sein (siehe S. 335 und S. 343).



# Kapitel 4

## Unterrichtsvorschläge und -konzepte zur Einbeziehung von Elementen der 3D-Computergrafik in das Stoffgebiet Analytische Geometrie

In dem vorliegenden Kapitel werden – auf der Grundlage der Sachanalyse in Kapitel 2 und der in den Kapiteln 1 und 3 angestellten didaktischen Überlegungen – Konzepte für die Einbeziehung ausgewählter Elemente der Computergrafik in den Unterricht beschrieben. Im Vordergrund stehen dabei Inhalte und Vorgehensweisen der Computergrafik, deren mathematische Grundlagen ohnehin Bestandteile der derzeitigen Curricula sind oder einen engen Bezug zu gegenwärtig behandelten Themengebieten aufweisen. Diesbezügliche Möglichkeiten aufzuzeigen ist notwendig, da geringe Chancen für eine Einbeziehung von Elementen der Computergrafik bestehen dürften, wenn damit gleichzeitig eine wesentliche Ausweitung der im Unterricht zu behandelnden mathematischen Inhalte erforderlich wird. Dennoch werden sich Hinweise auf wünschenswerte Modifikationen, zumindest jedoch Akzentuierungen auch der mathematischen Gegenstände des Stoffgebietes ergeben. Die unterbreiteten Vorschläge erfassen daher die Spannweite von unmittelbaren visuellen Ergänzungen der klassischen Inhalte des Stoffgebietes bis hin zu inhaltlichen Schwerpunktverschiebungen und der Einbeziehung von mathematischen Grundlagen der 3D-Computergrafik und -animation als Unterrichtsgegenstand.

Die Anordnung der Abschnitte dieses Kapitels gibt keine chronologische Gliederung des Unterrichtsstoffes vor – es wird ohnehin nicht möglich sein, alle hier unterbreiteten Vorschläge innerhalb eines Kurses umzusetzen, eine Auswahl ist in jedem Falle erforderlich. Mögliche Lehrgangsvarianten unter Einbeziehung der beschriebenen Konzepte werden am Ende des Kapitels in Abschnitt 4.9 diskutiert. Fundamentaler Bestandteil aller dort vorgeschlagenen Varianten ist die geometrische Modellierung von Objekten unter Verwendung räumlicher Koordinaten einschließlich der dabei erfolgenden Einarbeitung in eine geeignete Software. Dieser Gegenstandsbereich sollte den Einstieg in das Stoffgebiet Analytische Geometrie bilden, wenn beabsichtigt wird, Elemente der

3D-Computergrafik in nennenswertem Umfang einzubeziehen. Die meisten der in dem vorliegenden Kapitel dargelegten Vorschläge bauen daher auf den in Abschnitt 4.1 beschriebenen Einstieg auf.<sup>1</sup>

Die Darstellung der mathematischen Gegenstände ist bei den in diesem Kapitel dargestellten Konzepten ihrer Behandlung recht kurz gehalten, da diese bereits in Kapitel 2 beschrieben wurden. Während dort mathematische Inhalte anhand ihrer Anwendung für die grundlegenden Arbeitsschritte der Computergrafik zur Sprache kamen, ist das vorliegende Kapitel gewissermaßen umgekehrt strukturiert: Elemente der Computergrafik werden in die Behandlung mathematischer Themengebiete an Stellen einbezogen, an denen dafür geeignete Möglichkeiten bestehen. Dennoch finden sich die grundlegenden, in Kapitel 2 beschriebenen, Arbeitsschritte der 3D-Computergrafik (Modellierung, Transformationen, Rendering und Animation) hier wieder, was auf die bereits ausgeführte inhaltliche Nähe zwischen der 3D-Computergrafik und der analytischen Geometrie zurückzuführen ist. Somit treten bei den meisten der im Folgenden beschriebenen Vorschläge Werkzeug- und Gegenstandsaspekte (siehe Abschnitt 3.2.2, S. 189) der Einbeziehung der 3D-Computergrafik in den Unterricht miteinander kombiniert auf.<sup>2</sup>

Bei der Mehrzahl der im Folgenden unterbreiteten Vorschläge wurde besonderes Augenmerk auf ihre Umsetzbarkeit in Grundkursen gelegt – ein wichtiges Ziel der vorliegenden Arbeit besteht darin, Wege aufzuzeigen, Schüler für die Beschäftigung mit Mathematik zu motivieren und bei ihnen ein grundlegendes Verständnis mathematischer Arbeitsweisen zu entwickeln, die sich ansonsten nicht besonders für dieses Fach interessierten und eher dazu neigten, Aufgaben schematisch abzuarbeiten. Für Leistungskurse mit besonders interessierten und talentierten Schülern enthält bereits Kapitel 2 eine Fülle von Möglichkeiten, mathematische Grundlagen der Computergrafik im Unterricht zu thematisieren; in derartigen Kursen ist ein geringeres Maß an didaktischer Reduktion erforderlich als in Grundkursen. An einigen Stellen werden Erweiterungsmöglichkeiten der unterbreiteten Vorschläge bei der Umsetzung mit besonders leistungsstarken Schülern aufgezeigt. Diese beinhalten insbesondere anspruchsvolle Querverbindungen zum Stoffgebiet Analysis.

Obwohl Erfahrungen aus Unterrichtsversuchen schwerpunktmäßig erst in Kapitel 5 beschrieben werden, fließen sie an einigen Stellen des vorliegenden Kapitels bereits in die Überlegungen ein – eine strikte Trennung erscheint u. a. aus dem Grunde nicht sinnvoll, dass gesammelte Unterrichtserfahrungen zu Modifikationen ursprünglich geplanter Vorgehensweisen geführt und somit bereits auf einige der im Folgenden unterbreiteten Vorschläge zurückgewirkt haben.

---

<sup>1</sup>Eine Ausnahme bildet die in Abschnitt 4.2 beschriebene Nutzung einer Software für Visualisierungen von Inhalten der Vektorgeometrie, die prinzipiell auch ohne vorherige koordinatengeometrische Beschreibungen geometrischer Objekte erfolgen könnten. Dies entspräche allerdings aus vielen Gründen, die in den Kapiteln 1 und 3 dargelegt wurden, nicht den Intentionen der vorliegenden Arbeit.

<sup>2</sup>Ausnahmen bilden die Abschnitte 4.2 und (weitgehend) 4.5, in denen der Werkzeugcharakter einer 3D-Grafiksoftware oder eines CAS für visuelles und experimentelles Arbeiten im Vordergrund steht.

## 4.1 Geometrische Modellierung mittels räumlicher Koordinaten als Einstieg in das Stoffgebiet

### Herangehensweise und Ziele

Aus mehreren, bereits in Kapitel 1 dargestellten, Gründen erscheint es sinnvoll, zu Beginn des Stoffgebietes Analytische Geometrie mit räumlichen Koordinaten zu arbeiten und Objekte des Raumes durch Koordinaten sowie (zunächst in Ansätzen) Gleichungen zu beschreiben.<sup>3</sup> Dabei liegt es – auch aufgrund der Wirkungsweise der 3D-Computergrafik (vgl. Abschnitt 2.2) – nahe, den Einstieg in die räumliche Koordinatengeometrie mit der Modellierung einfacher Objekte in einer geeigneten Software (wie POV-Ray) zu verbinden, womit sich vor allem folgende Ziele verfolgen lassen:

- Die Schüler können das räumliche Koordinatensystem und die Lage von Punkten in Abhängigkeit von ihren Koordinaten veranschaulichen; sie erwerben in gewissem Maße eine „koordinatenbezogene Raumvorstellung“ (siehe Abschnitt 1.4.4).
- Ausgehend von elementargeometrischen Überlegungen lernen sie, einfache geometrische Körper durch Koordinaten bestimmender Punkte und charakterisierende Größen zu beschreiben sowie räumliche Szenen zu strukturieren.
- Die Schüler erlangen die Fähigkeit, die Anzahl der in Betracht zu ziehenden Dimensionen zu reduzieren und nutzen sie als Strategie der Vereinfachung von Positionierungsproblemen.
- Sie erkennen, dass es für die geeignete Positionierung von Objekten sinnvoll sein kann, Abstände zu berechnen und geometrische Objekte durch Gleichungen zu beschreiben. Anknüpfend an Unterrichtsinhalte der SI berechnen sie Abstände durch Koordinaten beschriebener Punkte und stellen Gleichungen für Kreise und Kugeln auf. Diese nutzen sie für die Lösung von Positionierungsproblemen.
- Der ästhetische Reiz selbst geschaffener computergrafischer Darstellungen motiviert die Schüler für die Beschäftigung mit analytischer Geometrie.
- Durch die Arbeit mit einer Grafiksoftware wird ein hohes Maß motivierender Schülertätigkeiten ermöglicht, anhand derer sich die Einführung neuer mathematischer Inhalte als notwendig und sinnvoll erweist.

#### 4.1.1 Beschreibung einfacher Körper durch Koordinaten

Anhand der Modellierung eines einfachen, aus geometrischen Grundkörpern zusammengesetzten „realen“ Objekts können die Schüler erste Fähigkeiten sowohl bei der Arbeit mit räumlichen Koordinaten als auch bei der Bedienung einer Grafiksoftware<sup>4</sup> erwerben.

---

<sup>3</sup>Der Übergang zu einer vektoriellen Raumgeometrie sollte möglichst dann erfolgen, wenn für die Schüler Grenzen der Leistungsfähigkeit von Koordinatendarstellungen sichtbar werden und sie hinreichend viele Beispiele kennengelernt haben, für die der Vektorbegriff eine Verallgemeinerung darstellt.

<sup>4</sup>Die folgenden Erläuterungen beziehen sich auf die Verwendung von POV-Ray. Wie in Abschnitt 3.4 ausgeführt wurde, könnten jedoch in analoger Weise auch andere Softwarepakete genutzt werden; ein Beispiel für die Verwendung des CAS MuPAD wird auf S. 230 angegeben.

Wie bereits in Abschnitt 3.5 ausgeführt wurde, ist es sinnvoll, den Schülern Vorlagen sowie eine Anleitung mit den benötigten Anweisungen (siehe S. 330) zur Verfügung zu stellen, anhand derer sie sich sofort auf die Beschreibung einfacher geometrischer Objekte durch Koordinaten konzentrieren können. Die Aufgaben, welche die Schüler unter Verwendung dieser Materialien bearbeiten, sollten folgenden Bedingungen genügen:

- Die Aufgaben sind für die Schüler interessant und herausfordernd.
- Es sind attraktive Ergebnisse mit recht wenigen Syntaxelementen möglich.
- Die Schwierigkeiten bei der Lösung der Aufgaben verlagern sich schnell von der Bedienung der Software hin zu Problemen der geometrischen Modellierung (Wahl geeigneter Formen, Anordnung von Objekten im Raum durch Koordinaten).

Ein Beispiel für eine als Einstieg gut geeignete Aufgabe, die diese Bedingungen erfüllt, den Schülern Gestaltungsfreiräume lässt und gleichzeitig Anforderungen stellt, die von allen Schülern erfüllt werden müssen (so dass eine gemeinsame Diskussion von Problemen und Ergebnissen erfolgen kann), lautet:

*Modellieren Sie einen Schneemann.<sup>5</sup>*

Da für den Bau eines Schneemannes lediglich Kugeln, Zylinder und ein Kegel benötigt werden, genügt für diese Aufgabe – bei Nutzung der in Abschnitt 3.5 beschriebenen Vorlagen – die Verwendung weniger Anweisungen von POV-Ray; die Schüler können sich somit schnell auf die Positionierung der Bestandteile des Schneemannes konzentrieren.

Um einen realistisch erscheinenden Schneemann zu modellieren, müssen die Schüler vor allem Überlegungen zur geeigneten Wahl der Mittelpunktsskordinaten und Radien der Komponenten (Rumpf, Kopf, Hut, Nase, Augen, Knöpfe) des Schneemannes anstellen. Dadurch erwerben sie erste Erfahrungen und ein „Gefühl“ für die Positionierung räumlicher Körper durch Koordinaten. Ohne Gleichungen für Kreise und Kugeln zu kennen, können sie sich durch Probieren und anschließendes Ändern der Koordinaten schrittweise an eine sinnvolle Anordnung „herantasten“ – dieser Weg wird in der Anfangsphase des Stoffgebietes Analytische Geometrie am häufigsten beschritten. Allerdings ist es dazu nötig, eine Ausgangsposition zu wählen, die in der „Nähe“ der gewünschten Position liegt (z. B. mittels Überlegungen zu den Koordinatenintervallen der die Kugeln begrenzenden Würfel). Durch geschicktes Variieren der Koordinaten mithilfe von Skizzen auf Papier gelang einigen Schülern sogar bereits eine annähernd realistische Anordnung von Knöpfen und Augen des Schneemannes. Abb. 4.1 zeigt erste Ergebnisse von Schülern nach ca. 30-40 Minuten Arbeit mit POV-Ray.<sup>6</sup>

<sup>5</sup>Eine Alternative oder Ergänzung bestünde in der Modellierung von Schachfiguren, die größtenteils ebenfalls mithilfe weniger Grundkörper möglich ist und zudem Kooperations- sowie Differenzierungsmöglichkeiten hinsichtlich der Schwierigkeitsgrade der verschiedenen Figuren bietet. Ein weiteres Beispiel ist die von LEUDERS in [159] beschriebene Modellierung eines stilisierten Elefanten.

<sup>6</sup>Die Abbildungen und zugehörigen Quelltexte fertigten Schüler eines Grundkurses ma-13 im Rahmen einer Unterrichtsreihe an, die in Abschnitt 5.1 beschrieben wird. Aus Platzgründen werden nur wenige Beispiele dargestellt, dabei aber sowohl recht fortgeschrittene als auch weniger ausgereifte Ergebnisse berücksichtigt, um einen Überblick über die Spannweite der Resultate zu vermitteln. Unter-



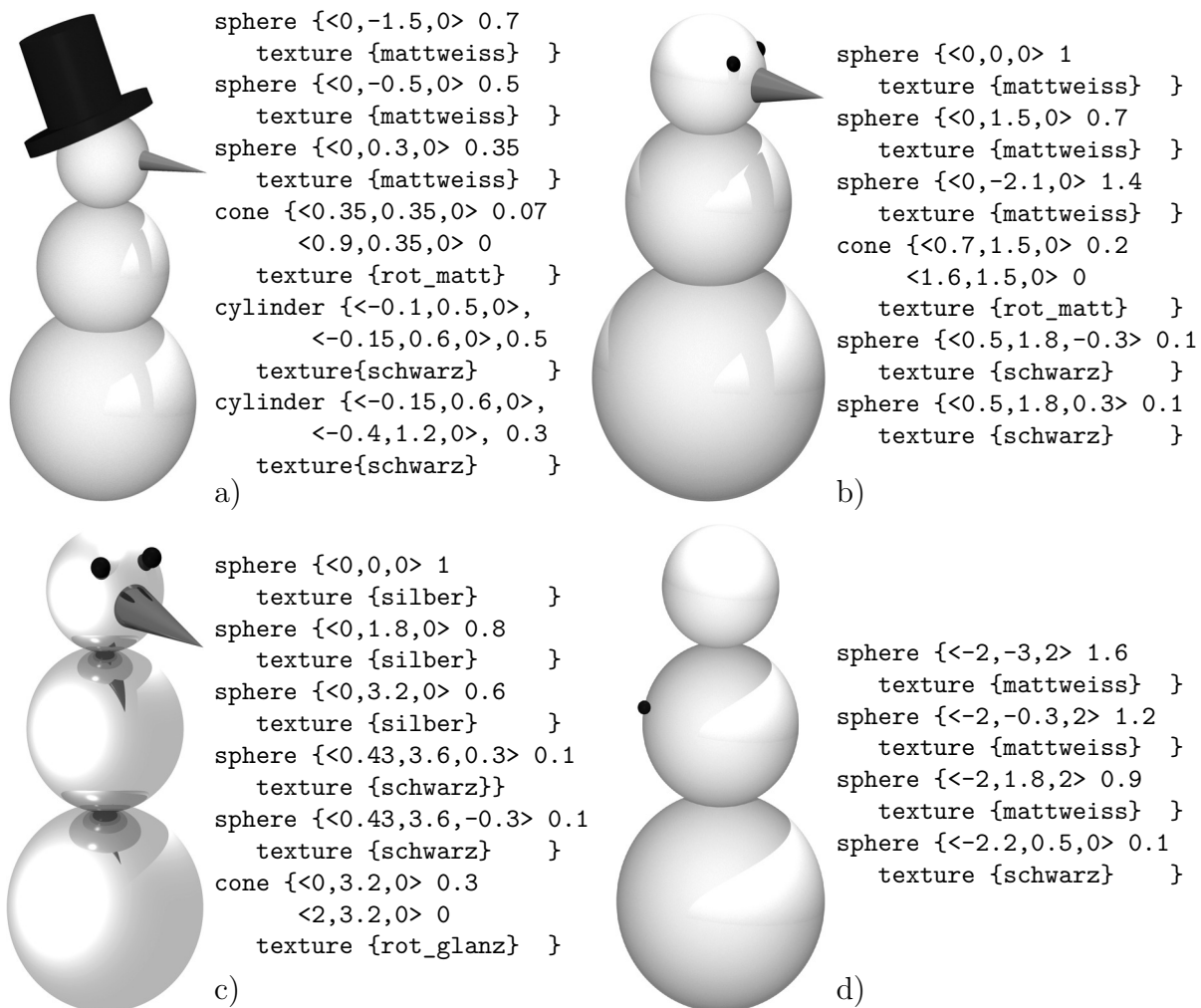


Abbildung 4.1: Ergebnisse von Schülern nach ca. 30-40 Minuten Arbeit mit POV-Ray

Bei den durchgeführten Unterrichtsversuchen zeigte sich, dass die Bedienung von POV-Ray (unter Nutzung der in Abschnitt 3.5 beschriebenen Anleitungen und Vorlagen) für die Schüler keine große Schwierigkeit darstellte. Nur in den ersten Minuten traten Probleme mit der Syntax der Skriptsprache auf (vergessene Klammern, Groß- und Kleinschreibung); schnell konnten die Schüler die entsprechenden Fehlermeldungen selbstständig berücksichtigen. Nach 20 Minuten hatten alle von ihnen mindestens Kopf und Rumpf des Schneemanns angeordnet, wobei einige Schüler bewusst Überlappungen, die beim Bau eines realen Schneemanns ebenfalls auftreten, modellierten. Größere Schwierigkeiten bereitete die sinnvolle und exakte Positionierung der Accessoires – Au-

schiede zwischen den dargestellten Schülerergebnissen sind vor allem hinsichtlich der Genauigkeiten von Positionierungen erkennbar. Es wird deutlich, dass für die Anordnung der Augen in dem in Abb. 4.1 b) dargestellten Beispiel bereits einige Zeit für Vorüberlegungen bzw. Versuche aufgewendet wurde. In den Ergebnissen einiger Schüler waren die Augen noch etwas weiter vom Kopf bzw. die „Knöpfe“ vom Bauch entfernt als in den Beispielen c) und d). Diese und weitere Dateien, welche Schüler angefertigt haben, können unter <http://www.afiller.de/habil> heruntergeladen werden (siehe Anhang D, S. 365).

gen und Knöpfe schwebten teilweise im Raum. Zum Teil zeigten sich diese Probleme erst, nachdem die Szenen aus anderen Blickwinkeln betrachtet wurden. Damit hatte sich die Problematik sehr schnell von technischen Fragen zur richtigen Beschreibung und Anordnung geometrischer Körper durch Koordinaten verlagert.

Wie erwartet wurde, näherten sich die Schüler durch schrittweises Verändern der Koordinaten der richtigen Position und Größe der Einzelteile an. Einige Schüler warfen dabei die Frage auf, wie es möglich ist, die Knöpfe exakt am Bauch und die Augen am Kopf des Schneemannes anzubringen. Die Beantwortung dieser Frage erfordert ein etwas tieferes Eindringen in die analytische Geometrie. Nachdem die Schüler erste Erfahrungen bei der geometrischen Modellierung in POV-Ray gesammelt haben, empfiehlt es sich daher, Vereinfachungsstrategien durch die Reduktion der Anzahl zu berücksichtigender Dimensionen zu thematisieren. Exakte Positionierungen werden durch die Beschreibung geometrischer Objekte (in dem hier betrachteten Falle Kugeln) durch Gleichungen ermöglicht, deren Herleitung in einem nächsten Schritt erfolgt.

#### 4.1.2 Reduktion der zu berücksichtigenden Dimensionen durch die Betrachtung von Schnitten mit Koordinatenebenen

Die Betrachtung von Schnitten zwischen Körpern bzw. Flächen und Ebenen gehört zu den wichtigsten Arbeitsweisen in der (elementaren wie analytischen) Raumgeometrie, da sie die Anwendung von Vorgehensweisen der ebenen Geometrie auf räumliche Sachverhalte ermöglicht. Vorstellungen von Schnitten sind zudem auch für die Beschreibung von Flächen durch Gleichungen von Bedeutung.<sup>7</sup>

In elementarer Weise können Schnittbetrachtungen bereits bei der Modellierung aus Grundkörpern zusammengesetzter einfacher Objekte thematisiert und genutzt werden. So erweist sich bei der auf Koordinatenbeschreibungen basierenden Konstruktion eines Schneemannes die sinnvolle Anordnung der Bestandteile entlang von Koordinatenachsen (sofern dies möglich ist) oder zumindest in Koordinatenebenen als wichtige Strategie der Vereinfachung von Positionierungsproblemen. In vielen Fällen lassen sich dadurch Probleme der Anordnung im dreidimensionalen Raum auf die Anordnung von Punkten auf einer Geraden oder in einer Ebene zurückführen. Schüler, welche die Mittelpunkte der „großen“ Kugeln des Schneemannes nicht auf einer Koordinatenachse positioniert hatten, bemerkten recht schnell, dass die Anordnung der weiteren Bestandteile, die sich an Kopf und Rumpf ausrichten müssen, dadurch erschwert wird.

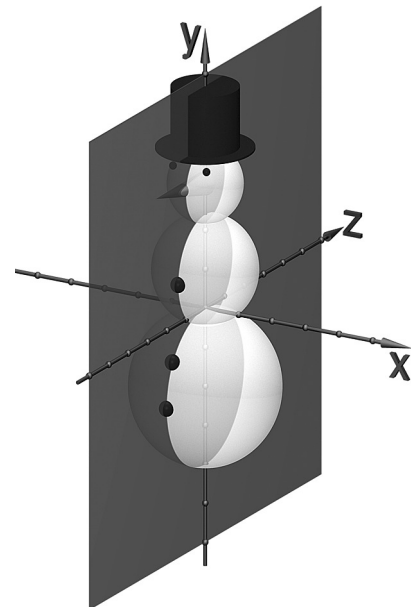


Abbildung 4.2: Schnitt eines Schneemannes mit einer Koordinatenebene

<sup>7</sup>Schnittbetrachtungen werden noch im Zusammenhang mit mehreren in dieser Arbeit unterbreiteten Vorschlägen als eine zentrale Strategie auftreten, siehe u. a. Abschnitt 4.5.

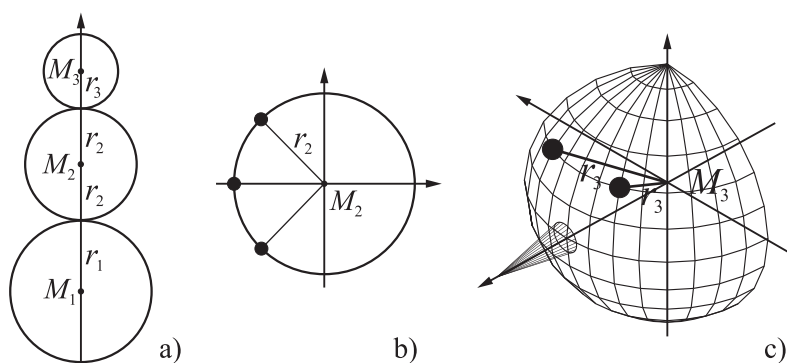


Abbildung 4.3: Positionierungsaufgaben in unterschiedlich vielen Dimensionen

Bei Betrachtung der Schnittfigur eines Schneemannes mit einer geeigneten Koordinatenebene (Abb. 4.2) fällt auf, dass bei einer günstigen Wahl der Koordinaten von Kopf und Rumpf drei Schwierigkeitsgrade hinsichtlich der exakten Berechnung der Positionen von Rumpf, Kopf, Knöpfen und Augen auftreten:

- Bei Anordnung der Hauptbestandteile des Schneemannes auf bzw. entlang einer Koordinatenachse müssen lediglich Mittelpunktskoordinaten und -radien sinnvoll addiert bzw. subtrahiert werden (siehe Abb. 4.3 a).
- Eine exakte Anordnung der Knöpfe auf dem Bauch kann durch zweidimensionale Überlegungen erreicht werden, erfordert aber die Erkenntnis, dass der Bauch eine geeignete Koordinatenebene in einem Kreis schneidet, dessen Radius mit dem der Kugel übereinstimmt (Großkreis). Mithilfe der Gleichung dieses Kreises können Koordinaten für die Mittelpunkte der Knöpfe ermittelt werden (Abb. 4.3 b).
- Die sinnvolle Anordnung der Augen auf einem Großkreis ist nicht möglich (vor allem dann nicht, wenn für die Nase eine möglichst einfache Position gewählt wurde). Die Positionierung der Augen erfordert somit dreidimensionale Überlegungen. Um die Mittelpunkte der (kugelförmigen) Augen exakt auf dem Kopf anzubringen, ist die Kugelgleichung des Kopfes zu betrachten (siehe Abb. 4.3 c).

### 4.1.3 Beschreibung einfacher geometrischer Figuren und Körper durch Gleichungen

Um Gleichungen für einfache geometrische Körper aufzustellen, lassen sich Unterrichtsinhalte der SI, insbesondere der Satz des Pythagoras, anwenden. Es ist leicht einsichtig, dass sich ebene Figuren einfacher durch Gleichungen beschreiben lassen als räumliche Körper bzw. Flächen. Gleichungen für Kreise können zudem hilfreich sein, um zu Kugelgleichungen (und auch zu Gleichungen für Zylinder und Kegel) zu gelangen.

Während die Herleitung der Kreisgleichung für viele Schüler in Leistungskursen kein großes Problem darstellt, bedarf es in Grundkursen erfahrungsgemäß einiger Hilfen und Impulse, um die Überlegungen der Schüler in eine erfolgreiche Richtung zu führen. Dazu können folgende Fragen zur Diskussion gestellt werden:

- Was ist ein Kreis?
- Wie lässt sich aus der Definition des Kreises eine Gleichung ableiten?
- Welcher spezielle Fall ist besonders einfach?

Nachdem von den Schülern herausgearbeitet wurde, dass ein Kreis in Mittelpunktslage die Menge aller Punkte ist, deren Abstände zum Koordinatenursprung gleich dem Radius des Kreises sind, können sie die Aufgabe erhalten, Abstände von Punkten zum Koordinatenursprung durch die Koordinaten der Punkte auszudrücken. Als zusätzliche Hilfe, um zu erkennen, dass hierzu der Satz des Pythagoras anzuwenden ist, kann eine Skizze wie Abb. 4.4 a) dienen. Mit dieser Hilfe bemerken erfahrungsgemäß viele Schüler, dass die Koordinaten eines Punktes des Kreises Längen der Katheten eines rechtwinkligen Dreiecks angeben, dessen Hypotenuse die Länge  $r$  hat. Somit gilt

$$x^2 + y^2 = r^2$$

für die Koordinaten aller Punkte des Kreises. Durch Verallgemeinerung der dazu durchgeführten Überlegungen und mithilfe einer Skizze wie z. B. in Abb. 4.4 b) dürfte auch die Aufstellung der Gleichung eines Kreises in allgemeiner Lage

$$(x - x_M)^2 + (y - y_M)^2 = r^2$$

den Schülern keine großen Schwierigkeiten bereiten.

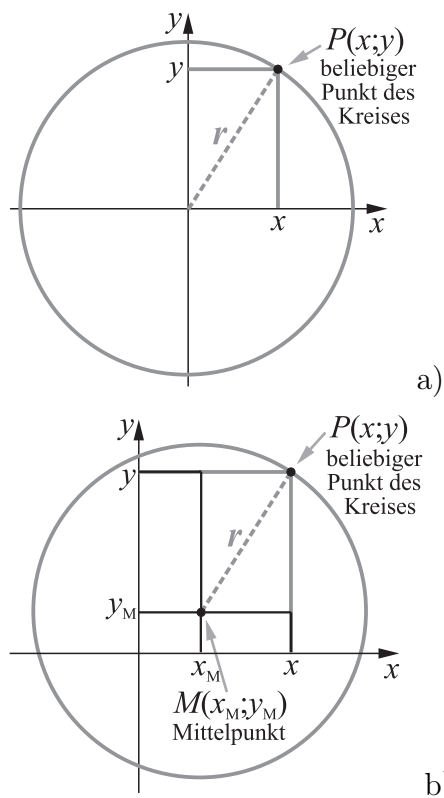


Abbildung 4.4: Skizzen zur Herleitung der Kreisgleichung

Nach der Herleitung der Kreisgleichungen in Mittelpunkts- und in allgemeiner Lage sowie einer kurzen Diskussion der Tatsache, dass es sich bei einer Kugel um eine Menge von Punkten handelt, die denselben Abstand vom Mittelpunkt haben, ist zu erwarten, dass viele Schüler aufgrund von (zunächst formalen) Analogieüberlegungen Vorschläge für die Gleichungen einer Kugel in Mittelpunkts- bzw. in allgemeiner Lage unterbreiten:

$$x^2 + y^2 + z^2 = r^2$$

bzw.

$$(x - x_M)^2 + (y - y_M)^2 + (z - z_M)^2 = r^2.$$

Zur Begründung sollten Überlegungen zur Länge der Raumdiagonalen eines Quaders angestellt werden, falls diese den Schülern nicht aus dem Unterricht der S I erinnerlich sind. Abb. 4.5 kann als Hinweis dienen, Kugelgleichungen auf die Längen der Raumdiagonalen von Quadern zurückzuführen.

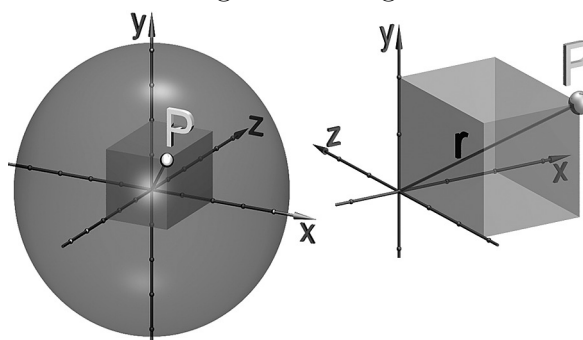


Abbildung 4.5: Unterstützende Grafiken zur Herleitung der Kugelgleichung

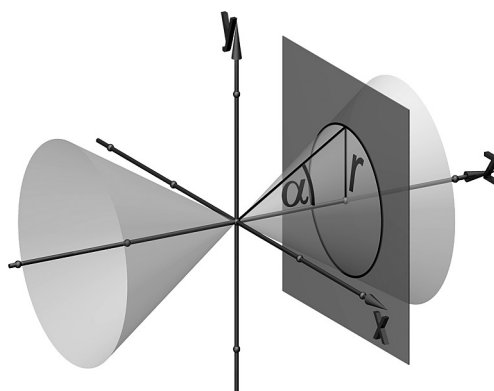
Bei der Herleitung der Kugelgleichung in allgemeiner Lage sollte den Schülern bewusst werden, dass sie damit gleichzeitig eine Möglichkeit gefunden haben, Abstände beliebiger Punkte  $P_1(x_1; y_1; z_1)$ ,  $P_2(x_2; y_2; z_2)$  im Raum aus deren Koordinaten zu berechnen:

$$|P_1P_2| = \sqrt{(x_2-x_1)^2 + (y_2-y_1)^2 + (z_2-z_1)^2}.$$

Als „Ausblick“ kann mitgeteilt werden, dass die Berechnung von Winkeln i. Allg. nicht auf eine derart elementare Weise möglich ist, sondern weiter gehende Überlegungen erfordert.

Nachdem die Schüler anhand der Kugel erste Erfahrungen bei der Beschreibung geometrischer Körper durch Gleichungen gesammelt haben, liegt es nahe, dass sie in weitgehend selbstständiger Arbeit die *Gleichung eines Kegels* herleiten.<sup>8</sup> Folgende Überlegungen sind dazu anzustellen:

- Wahl einer Lage des Kegels im Koordinatensystem,<sup>9</sup> welche die Herleitung einer Gleichung vereinfacht,<sup>10</sup>
- Betrachtung von Schnittkreisen des Kegels mit zur Kegelachse senkrechten Ebenen (im einfachsten Falle mit dem Abstand 1 zum Koordinatenursprung),
- Ermittlung des Radius eines Schnittkreises in Abhängigkeit vom Abstand der Ebene, in welcher der Kreis liegt, zur Kegelspitze sowie vom Öffnungswinkel  $2\alpha$  des Kegels,
- Nutzung der Gleichung des betrachteten Schnittkreises.



Durch diese Überlegungen ergibt sich (bei der in Abb. 4.6 dargestellten Lage des Kegels bezüglich des Koordinatensystems) die Gleichung

$$x^2 + y^2 = r^2 = (z \cdot \tan \alpha)^2 \quad \text{bzw.} \quad x^2 + y^2 - z^2 \tan^2 \alpha = 0$$

Abbildung 4.6: Herleitung der Gleichung eines Kegels in spezieller Lage

für den Kegel. Eine mögliche Vereinfachung für leistungsschwächere Schüler bestünde in der Betrachtung eines Kegels mit einem Öffnungswinkel von  $45^\circ$ .<sup>11</sup>

Die für die Herleitung der Kreis-, Kugel- und Kegelgleichungen benötigte Zeit hängt in starkem Maße davon ab, wie intensiv sich die Schüler in der SI mit dem Satz des Pythagoras und seinen Anwendungen beschäftigt haben. Evtl. haben sie dort bereits

<sup>8</sup>Zwar werden Gleichungen für Kegel im Zusammenhang mit dem Ausgangspunkt der Überlegungen zu Kreis- und Kugelgleichungen (die Positionierung der Bestandteile eines Schneemannes) nicht benötigt. Allerdings dürfte es für die Schüler leicht einsichtig sein, dass diese für Positionierungsaufgaben bei der Modellierung anderer Objekte nützlich sind. Auf die Kegelgleichungen kann im weiteren Verlauf des Unterrichts noch zurückgegriffen werden, siehe Abschnitt 4.5.1.

<sup>9</sup>Natürlich kann diese Überlegung auch dahin gehend geführt werden, ein geeignetes Koordinatensystem zu wählen, anhand dessen sich ein Kegel leicht durch eine Gleichung beschreiben lässt. Allerdings dürfte diese Herangehensweise für Schüler, welche i. Allg. keine Erfahrungen hinsichtlich der Arbeit mit verschiedenen Koordinatensystemen haben, eher fern liegen.

<sup>10</sup>Es wird für die Schüler nach den Überlegungen zum Aufbau des Schneemannes nahe liegen, die Kegelspitze im Koordinatenursprung und seine Achse entlang einer Koordinatenachse anzuordnen.

<sup>11</sup>In diesem Spezialfall gilt  $r = z$  und die Nutzung der trigonometrischen Beziehung  $\tan \alpha = \frac{r}{z}$  in dem in Abb. 4.6 dargestellten Dreieck ist nicht notwendig.

Gleichungen für den Kreis und u. U. sogar die Kugel hergeleitet. Falls bei den Schülern kaum Vorkenntnisse in Erinnerung sind, so lohnt es sich m. E., Zeit für eine an elementargeometrische Überlegungen anknüpfende metrische Koordinatengeometrie in dem hier kurz beschriebenen Sinne aufzuwenden<sup>12</sup> – ohne die Fähigkeit, derartige elementare Überlegungen zu führen, dürfte auch eine metrische analytische Geometrie auf der Grundlage des Skalarproduktes von vielen Schülern lediglich formal nachvollzogen werden.

#### 4.1.4 Anwendung von Kreis- und Kugelgleichungen für die Lösung von Positionierungsaufgaben

Nach der Herleitung von Gleichungen für Kreise und Kugeln können die Schüler diese nutzen, um zu exakten Anordnungen der Bestandteile ihrer ersten 3D-Szenen (wie der hier betrachteten Schneemänner) zu gelangen. Dazu ist evtl. noch kurz zu besprechen, dass bestimmte Koordinaten festzulegen sind, um die fehlenden Koordinaten daraus zu berechnen. Es ist natürlich sinnvoll, u. a. die Knöpfe in einer Koordinatenebene anzubringen. Wird dafür (wie in Abb. 4.2 auf S. 220) die  $y$ - $z$ -Ebene gewählt, so gilt für die Mittelpunkte der Knöpfe jeweils  $x = 0$ . Weiterhin sind für die Knöpfe Höhen ober- bzw. unterhalb des Mittelpunktes der Kugel festzulegen, deren Beträge kleiner als der Radius der Kugel sein müssen (siehe Abb. 4.7 a).

Es ergibt sich durch Einsetzen in die Kreisgleichung für die  $z$ -Koordinate eines Knopfes  $z^2 = r^2 - h^2$ ; je nachdem, wo sich die Vorderseite des Schneemannes befindet, ist die positive oder negative Lösung zu wählen. Auf analoge Weise kann der notwendige Radius eines Hutzyllinders in Abhängigkeit von der gewünschten Höhe über dem Mittelpunkt des Kopfes (oder umgekehrt die Höhe in Abhängigkeit vom Radius) berechnet werden (Abb. 4.7 b).

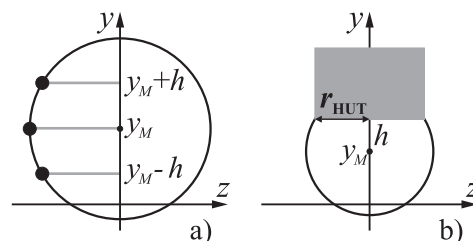


Abbildung 4.7: Positionierung auf einem Kreis

Alternativ zur Verwendung der Kreisgleichung könnte die Positionierung der Knöpfe auch durch die Anwendung trigonometrischer Beziehungen erfolgen. Dabei wären die Winkel zwischen den Verbindungsstrecken des Bauchmittelpunktes mit den Knopfmittelpunkten sowie der „Äquatorebene“ des Bauches vorzugeben, siehe Abb. 4.3 b) auf S. 221. Die Verallgemeinerung eines derartigen Vorgehens auf dreidimensionale Positionierungsaufgaben (wie z. B. der Augen) erfordert jedoch zeitlich aufwändigere

<sup>12</sup>Einen Weg, wie Schüler zu einer Gleichung für die Berechnung der Länge der Raumdiagonalen eines Quaders aus seinen Seitenlängen gelangen können, beschrieb POLYA ausführlich in [199] unter dem Aspekt des Problemlösens. Obwohl es sich dabei um einen Unterrichtsgegenstand der SI handelt, ist die Einbeziehung derartiger Überlegungen in den Unterricht der analytischen Geometrie m. E. notwendig, falls die Schüler nicht hinreichend damit vertraut sind. Natürlich zwingt die Relation zwischen den Rahmenplanforderungen und der zur Verfügung stehenden Zeit dabei zu Kompromissen. Die in einigen neueren Rahmenplänen zu beobachtende Tendenz weniger detaillierter Vorgaben hinsichtlich zu behandelnder Aufgabentypen (vgl. Abschnitt 1.3) könnte größere Spielräume schaffen, die Vorkenntnisse der Schüler angemessen zu berücksichtigen.

Betrachtungen, die der Umwandlung von Kugel- in kartesische Koordinaten entsprechen. Insbesondere zu Beginn des Stoffgebietes Analytische Geometrie ist deshalb m. E. die Nutzung der Kreisgleichung wegen der recht einfachen Übertragbarkeit des Vorgehens auf dreidimensionale Positionierungsaufgaben zu bevorzugen.

Für die Anbringung der Augen ist, wie bereits erwähnt wurde, die Nutzung der Kugelgleichung erforderlich. Dazu sind (bei Positionierung des Kopfmittelpunktes auf der  $y$ -Achse) die Höhe  $h_A = y_A - y_M$  der Augen über dem „Kopfäquator“ sowie ihre Abstände  $x_A$  von der  $y$ - $z$ -Ebene vorzugeben. Durch Einsetzen in die Kugelgleichung kann die  $z$ -Koordinate (die für beide Augen gleich ist, siehe Abb. 4.8) ermittelt werden:

$$z_A^2 = r^2 - (y_A - y_M)^2 - x_A^2.$$

Wiederum ist, je nach der Lage der Vorderseite des Schneemannes, die positive oder negative Lösung zu verwenden.

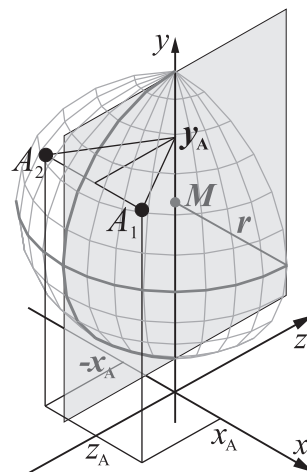


Abbildung 4.8: Positionierung auf einer Kugel

#### 4.1.5 Fertigstellung der ersten Grafiken durch die Schüler; Verwendung von Variablen, Berechnungen in POV-Ray

Die Berechnungen der exakten Positionen der Bestandteile des Schneemannes können die Schüler im Zusammenhang mit der weiteren Arbeit an ihren Grafiken durchführen. Nachdem sie einige Werte anhand konkreter Zahlen berechnet haben, empfiehlt es sich, auf zwei Aspekte der Beschreibungssprache von POV-Ray hinzuweisen, die bei komplexeren Szenen Erleichterungen schaffen und die Übersichtlichkeit verbessern können:<sup>13</sup>

- Die `#declare`-Anweisung ermöglicht die Festlegung von Variablen und die Zuweisung von Werten. Die so definierten Variablen können innerhalb der gesamten Szene genutzt werden.<sup>14</sup> Die Schüler sollten darauf hingewiesen werden, Variable so zu benennen, dass sich ihre Bedeutungen aus den Namen erschließen.
- Berechnungen kann POV-Ray selbst ausführen; es stehen neben den Grundrechenoperationen  $+$ ,  $-$ ,  $*$  und  $/$  auch Wurzeln, Exponential-, Logarithmus- und trigonometrische Funktionen sowie eine Vielzahl weiterer Funktionen zur Verfügung.<sup>15</sup>

Als Beispiel für die Definition von Variablen und die Nutzung der Rechenoperationen in POV-Ray seien die Beschreibungen des Kopfes und der Augen eines Schneemannes

<sup>13</sup>Die Nutzung dieser Sprachelemente entspricht dem Übergang vom Rechnen mit Zahlen zur Verwendung von Variablen. Auch bei der Arbeit mit CAS (und verstärkt beim Programmieren) sollte der Übergang von Berechnungen unter jeweiliger Eingabe konkreter Zahlen (Taschenrechnermodus) zur Definition von Variablen vollzogen werden.

<sup>14</sup>Eine Ausnahme bilden lokale Variable, die nur innerhalb von Prozeduren bzw. Makros gültig sind. Diese sind jedoch in dem hier betrachteten Zusammenhang von untergeordneter Bedeutung.

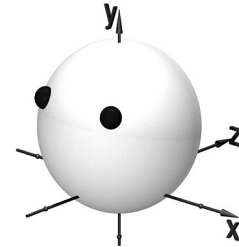
<sup>15</sup>Die in POV-Ray nutzbaren reellwertigen Funktionen sind in der Hilfe des Programms (Stichwort `float → functions`) bzw. in [200], S. 38–41 beschrieben.

angegeben, die bei vorheriger Berechnung der Werte der  $z$ -Koordinaten der Augen durch folgende Anweisungen beschrieben werden können:

```
sphere{<0, 1, 0> 1.5 texture{mattweiss}} // Kopf
sphere{<0.8, 1.6, -1.118> 0.2 texture{schwarz}} // Auge 1
sphere{<-0.8, 1.6, -1.118> 0.2 texture{schwarz}} // Auge 2
```

Unter Verwendung von Variablenamen und der internen Berechnungsmöglichkeiten von POV-Ray entsteht dieselbe Szene durch folgende Beschreibung:

```
#declare kopfhoehe = 1;
#declare kopfradius = 1.5;
#declare aughoehe = 0.6;
#declare xauge = 0.8;
#declare augradius = 0.2;
#declare zauge = - sqrt( pow(kopfradius,2) - pow(aughoehe,2) - pow(xauge,2) );
sphere{ <0, kopfhoehe, 0> kopfradius texture{mattweiss} } // Kopf
sphere{ < xauge, kopfhoehe+aughoehe, zauge> augradius texture{schwarz} } // Auge 1
sphere{ <-xauge, kopfhoehe+aughoehe, zauge> augradius texture{schwarz} } // Auge 2
```



Der Quelltext einer Szene ist, wie anhand dieses Beispiels sichtbar wird, unter Verwendung von Variablenamen und bei Durchführung der notwendigen Berechnungen innerhalb von POV-Ray länger als bei unmittelbarer Eingabe numerischer Werte,<sup>16</sup> weshalb viele Schüler zögern, in POV-Ray (und auch in CAS) mit Variablen zu arbeiten.<sup>17</sup> Der Vorteil dieser Arbeitsweise wird aber spätestens dann sichtbar, wenn „Gestaltungsexperimente“ durchgeführt werden sollen. So erfordert z. B. die Veränderung der Augenhöhe (bei fortbestehender Anordnung am Kopf) innerhalb der unteren Beschreibung nur die Änderung des Wertes einer einzigen Variablen, während in der obigen Variante mehrere Zahlenwerte verändert und z. T. neu berechnet werden müssen. Es kann auch sinnvoll sein, für Texturen Variablenamen festzulegen, falls mehrere Objekte dieselbe Textur erhalten und daran evtl. Veränderungen vorgenommen werden sollen.<sup>18</sup>

<sup>16</sup>Allerdings musste für die „kurze“ Version vorher der  $z$ -Koordinatenwert der Augen berechnet werden, was den durch die geringere Zahl an Eingabezeilen gegebenen Zeitvorteil relativiert.

<sup>17</sup>Im Unterricht und auch in Seminaren fällt es erfahrungsgemäß schwer, Schüler und Studierende von der Verwendung von Variablen überzeugen. Oft gelingt dies erst nach einer etwas längeren Einarbeitung in die Software. In jedem Falle erscheint es sinnvoll, die Vorteile einer strukturierten Gestaltung von Dateien durch die Verwendung von Variablen mehrfach zu thematisieren und auch anhand von Beispielen wie der erwähnten Änderung der Augenhöhe zu belegen. Grundsätzlich ist festzustellen, dass viele Anwender von Software recht wenig strukturiert arbeiten. So werden bei der Verwendung von Textverarbeitungen wie Word häufig physische Textauszeichnungen (fett, kursiv usw.) vorgenommen und verhältnismäßig selten Formatvorlagen (die als Variable für Textformatierungen aufgefasst werden können) genutzt; daraus resultiert oft ein inkonsistentes Erscheinungsbild von Dokumenten.

<sup>18</sup>Prinzipiell lassen sich Zahlen, Vektoren, geometrische und beliebige andere Objekte in POV-Ray als Variable definieren. Dieses Variablenkonzept ist somit allgemeiner als das im Mathematikunterricht übliche und entspricht der Variablennutzung in der Informatik. Das Arbeiten mit Variablen im Mathematikunterricht kann aber durch die Einbeziehung dieser Auffassung unterstützt werden, wenn für die Schüler anhand konkreter Beispiele erkennbar wird, dass durch die Verwendung von Variablen Vorteile hinsichtlich der Flexibilität von Berechnungen und Konstruktionen entstehen und sie Überlegungen zur sinnvollen Festlegung von Größen als Variable anstellen.



### 4.1.6 Fazit, Anmerkungen, Erfahrungen und einige Ergebnisse

Ein elementargeometrisch orientierter und mit der Modellierung in einer Grafiksoftware verbundener Einstieg in die analytische Geometrie, wie er hier skizziert wurde, kann dazu beitragen, die Nützlichkeit analytischer Beschreibungen geometrischer Objekte für Schüler „erlebbar“ zu machen und sie somit für das Stoffgebiet zu motivieren. Er vermeidet zudem die allzu frühe Einführung von abstrahierenden Objekten (insbesondere Vektoren), deren Bedeutung und Leistungsfähigkeit den Schülern zu Beginn des Stoffgebietes aufgrund einer nicht ausreichenden Abstraktionsbasis schwerlich bewusst werden kann. Die genannten Bezüge zu Unterrichtsinhalten der S I (Eigenschaften geometrischer Körper, Satz des Pythagoras) führen zu vertikalen Vernetzungen (vgl. 1.4.7, S. 51f.), die dazu beitragen können, den mitunter beklagten Eindruck der Schüler von der analytischen Geometrie als „isolierter Teildisziplin“ der Mathematik nicht oder zumindest in weniger starkem Maße aufkommen zu lassen.

Die Herangehensweise, geometrische Objekte nach einer Diskussion ihrer definierenden Eigenschaften zunächst durch Koordinaten bestimmender Punkte sowie charakteristische Größen zu beschreiben<sup>19</sup> und erst nach einigen diesbezüglichen Anwendungen Gleichungen aufzustellen, entspricht einer natürlichen Betrachtungsweise, die von ganzheitlichen Aspekten ausgeht und erst in der Folge analytische Beschreibungen entwickelt. Der Entwicklung eines Verständnisses geometrischer Objekte als Punktmengen kann mit den in diesem Abschnitt geschilderten Überlegungen allerdings nicht hinreichend Rechnung getragen werden – dieser Gedanke liegt der Beschreibung durch Gleichungen zwar zugrunde, verankert sich dabei aber erfahrungsgemäß nur ansatzweise in den Vorstellungen der Schüler. Für die Ausprägung des „Punktmengengedankens“ bestehen vor allem bei der Beschäftigung mit Parameterdarstellungen gute Möglichkeiten (siehe 3.3.2), auf die in Abschnitt 4.3 noch eingegangen wird.

Die vorgeschlagene Behandlung von Kreisen und Kugeln vor der Behandlung von Geraden und Ebenen ist für einen Kurs zur analytischen Geometrie ungewöhnlich. Diese Reihenfolge ergibt sich aus der Anknüpfung an elementargeometrische Überlegungen und aus dem Ansatz der Modellierung „realer“ (und somit endlich ausgedehnter) Objekte.<sup>20</sup> Aus didaktischer Sicht bestehen m. E. keine größeren Schwierigkeiten, wenn Schüler die Beschreibung von Objekten im Raum durch Gleichungen erstmals anhand von Kugeln anstelle von Ebenen vornehmen. Aus fachsystematischer Sicht könnte dagegen der Einwand erhoben werden, dass die frühe Berücksichtigung metrischer Eigenschaften (welche bei der Behandlung von Kugeln unabdingbar ist) die Trennung zwischen affinen und metrischen Aspekten der analytischen Geometrie erschwert bzw. sogar unmöglich macht. Allerdings wurde bereits in Kapitel 1 deutlich, dass ein tiefer gehendes Verständnis für die Unterschiede zwischen affiner und metrischer Geometrie in der Schule

<sup>19</sup>Diese Art der Beschreibung weist auch Bezüge zur Konstruktion von Körpern bzw. dazu analoger ebener Figuren auf, siehe die diesbezüglichen Ausführungen auf S. 51f.

<sup>20</sup>Für einen nichtgeometrischen Anwendungen betonenden Unterricht der linearen Algebra ist der hier skizzierte Einstieg allerdings ungeeignet, da bei dieser Zielrichtung Geraden und Ebenen primär Lösungsmengen linearer Gleichungen bzw. Gleichungssysteme veranschaulichen sollen.

ohnehin kaum zu erreichen ist. Da die überwiegende Zahl der Schüler aufgrund ihrer Vorkenntnisse hierfür über keine ausreichende Abstraktionsbasis verfügt, würde eine solche Trennung eher als gekünstelt empfunden werden. Die meisten Rahmenpläne fordern eine konsequente Trennung zwischen affinen und metrischen Aspekten nicht mehr. Aus diesen Gründen impliziert die frühe Einbeziehung metrischer Eigenschaften m. E. keinen nennenswerten Nachteil des hier skizzierten Vorschlags.

In einem Unterrichtsversuch, der unter recht starken zeitlichen Beschränkungen durchgeführt wurde (vgl. 5.1), nutzten nicht alle Schüler die hergeleiteten Kreis- und Kugelgleichungen für die Positionierung der Knöpfe und Augen; insbesondere diejenigen, die vorher bereits eine zufrieden stellende Anordnung durch versuchsweises „Herantasten“ erreicht hatten, sahen dafür keine Notwendigkeit mehr.<sup>21</sup> Jedoch wurde m. E. auch für diese Schüler plausibel, dass die Nutzung der Gleichungen bei komplizierteren Positionierungsaufgaben sinnvoll sein kann. In jedem Falle haben die Schüler durch die Arbeit an der sinnvollen Anordnung einiger Körper, auch durch schrittweises Annähern, eine gewisse „Orientierungsfähigkeit“ im räumlichen Koordinatensystem erlangt, die für das gesamte Stoffgebiet Analytische Geometrie und darüber hinaus von Nutzen ist.

Der hier vorgeschlagene Einstieg in das Stoffgebiet Analytische Geometrie erfordert natürlich Unterrichtszeit, wobei der Aufwand insbesondere für die Beschreibung von Kreisen und Kugeln durch Gleichungen stark von den Vorkenntnissen der Schüler aus der SI abhängt (siehe die entsprechenden Bemerkungen auf S. 223f.). Abstriche an den traditionellen, an algebraischen Strukturen orientierten Inhalten des Stoffgebietes werden somit insbesondere in Grundkursen notwendig. Das Setzen von Schwerpunkten, die vertieft sowie durch Anwendungen motiviert und gefestigt werden, sowie die Verringerung des Anteils an Routineaufgaben<sup>22</sup> sind Ansätze für notwendige Straffungen, die zugleich geometrische Aspekte stärker in den Mittelpunkt stellen.<sup>23</sup>

Zu differenzieren ist zwischen der für mathematische Überlegungen zur Beschreibung geometrischer Objekte aufgewendeten und der für mehr technische Aspekte des Umgangs mit der Software benötigten Zeit – auch wenn diese beiden Bereiche oft gleichzeitig von Bedeutung sind. Natürlich kann im Mathematikunterricht nicht sehr viel Unterrichtszeit für die Arbeit der Schüler an ihren Grafiken aufgewendet werden. Insgesamt reichen dafür erfahrungsgemäß aber (als Minimum) drei Unterrichtsstunden aus, näm-

---

<sup>21</sup>Die beabsichtigte Motivierung der Kugelgleichung durch ihre Notwendigkeit für exakte Positionierungen gelang somit nicht vollständig überzeugend. Allerdings war dies m. E. auch durch die Knappheit der Zeit bedingt; anschließende weitere Beispiele und die Thematisierung der POV-Ray-internen Berechnungsmöglichkeiten hätten die Chance erhöht, dass Schüler Gleichungen geometrischer Objekte für Positionierungsaufgaben nutzen. Entsprechende positive Erfahrungen konnten allerdings nur – unter geringerem Zeitdruck – in Seminaren mit Studierenden gesammelt werden.

<sup>22</sup>Das durch visuelle Vorstellungen unterstützte Verständnis kann die Reduktion der für Routineaufgaben aufgewendeten Zeit zumindest zu einem gewissen Teil kompensieren.

<sup>23</sup>Es soll nicht verschwiegen werden, dass der Zeitbedarf für die in diesem Abschnitt skizzierten Überlegungen und die Arbeit mit der Software bei Curricula mit sehr detaillierten Vorgaben zur Behandlung von Aufgabentypen bereits problematisch sein kann. Auf diese Problematik wird bei der Beschreibung eines Unterrichtsversuches in Abschnitt 5.1 noch zurückzukommen sein.

lich eine Stunde für die in Abschnitt 4.1.1 beschriebenen ersten Versuche und – nach Durchführung der beschriebenen Schnittbetrachtungen sowie der Herleitung der Kreis- und Kugelgleichungen – zwei Stunden für die Perfektionierung der Grafiken.<sup>24</sup> Für die Anfertigung anspruchsvoller Darstellungen benötigen Schüler i. Allg. wesentlich mehr Zeit; einen großen Teil der Arbeit können sie jedoch außerhalb der Unterrichtszeit verrichten – viele Schüler verbringen aus Interesse und Freude an der Anfertigung einer 3D-Computergrafik freiwillig recht viel Zeit in häuslicher Arbeit damit. Als Beispiel hierfür sei der von einer Schülerin eines Grundkurses folgendermaßen beschriebene, in Abb. 4.9 in zwei Ansichten dargestellte, Schneemann angegeben:

```
sphere { <2,-1.2,-2> 1.6 texture { mattweiss } }           //unterkörper
sphere { <2,1.2,-2> 1.2 texture { mattweiss } }           //oberkörper
sphere { <2,3,-2> 0.8 texture { mattweiss } }             //kopf
cylinder { <2,3.5,-2>,<2,3.55,-2> 1.0}                   //zylinderplatte
cylinder { <2,3.55,-2>,<2,3.7,-2> 0.6}                   //zylinderkegel
cylinder { <2,3.7,-2>,<2,3.8,-2> 0.6 texture { rot_matt } } //bandole
cylinder { <2,3.8,-2>,<2,4.3,-2> 0.6}                   //zylinderkegel
cone { <1.7,3,-1.8>0.3,<3.2,3,-3>0 texture { rot_matt } } //nase
sphere { <2.35,3.2,-2.7>0.09 }                           //auge links
sphere { <2.7,3.2,-2.3>0.09 }                           //auge rechts
cone { <1.5,1.65,-1.85> 0.5, <0,1.3,-3.45> 0.3 texture {mattweiss} } //arm links
sphere { <0,1.3,-3.45> 0.38 texture { mattweiss } }       //hand links
cone { <2.6,1.65,-1.8> 0.5, <4.3,1.3,-1.8> 0.3 texture {mattweiss} } //arm rechts
sphere { <4.3,1.3,-1.8> 0.38 texture {mattweiss} }       //hand rechts
cylinder { <-0.2,2.3,-3.55>, <0.8,-2.8,-2.4> 0.1 texture {holz} } //besenstiel
box { <0.8,2.3,-3.15>, <-1.1,2.5,-3.85> texture { holz } } //besenkopf
```

<sup>24</sup>Nach dieser Zeit waren in Unterrichtsversuchen alle Schüler in der Lage, POV-Ray für die Erstellung einfacher Grafiken zu nutzen und hatten einen Schneemann (mit sehr unterschiedlichem Grad an Perfektion und teilweise einigen verbleibenden unkorrekten Positionierungen) erstellt. Allerdings ist das Minimum von drei Stunden bei weitem nicht ausreichend, um Schüler zu einer strukturierten Arbeitsweise, insbesondere durch die Verwendung von Variablendeklarationen (siehe S. 225) zu führen. Auf diesen Aspekt sollte bei der Nutzung der Software im weiteren Unterricht anhand von Beispielen zurückgekommen werden (siehe z. B. Abschnitt 4.2).

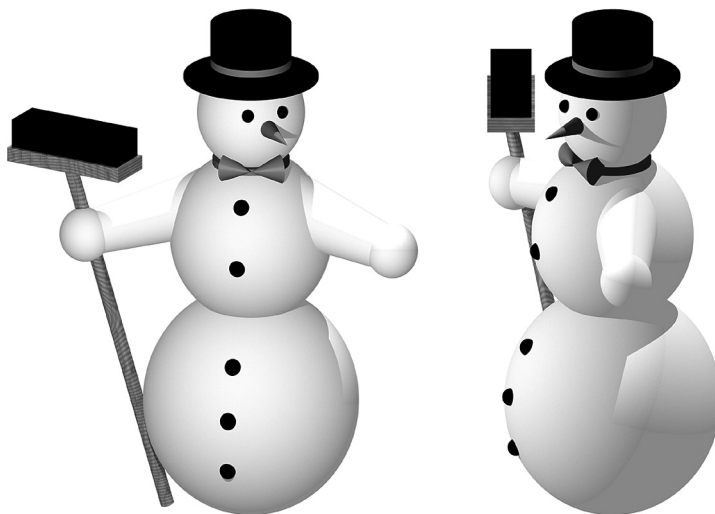


Abbildung 4.9: Grafik einer Schülerin

Farbige Versionen dieser und weiterer von Schülern eines Grundkurses ma-13 sowie einer 12. Klasse zu Beginn des Stoffgebietes Analytische Geometrie angefertigte Grafiken enthält Anhang B auf S. 357ff.; die zugehörigen POV-Ray-Beschreibungen befinden sich in dem Ordner *Schuelerarbeiten* (siehe Anhang D, S. 365).

```

box { <0.75,2.5,-3.2>, <-1.05,2.9,-3.8> } //besenborsten
cylinder { <2,2.25,-2>, <2,2.35,-2> 0.57 texture { rot_matt } } //kragenschleife
cone { <1.9,2.35,-2.6>0.2 <2.34,2.35,-2.6>0 texture {rot_matt} } //schleife li.
cone { <2.65,2.35,-2.25>0.2 <2.34,2.35,-2.6>0 texture {rot_matt} } //schleife re.
sphere { <2.34,1.9,-2.87>0.12 } //oberster knopf
sphere { <2.4,1.1,-3.1>0.12 } //zweiter knopf
sphere { <2.41,-0.3,-3.25>0.12 } //dritter knopf
sphere { <2.48,-1,-3.51>0.12 } //vierter knopf
sphere { <2.43,-1.75,-3.43>0.12 } //fünfter knopf

```

#### 4.1.7 Verwendung eines CAS für den beschriebenen Einstieg

Obwohl sich die Nutzung einer fotorealistischen Grafiksoftware wie POV-Ray für die Modellierung der Realität nachempfundener Objekte besonders anbietet, sind die beschriebenen Vorgehensweisen auch unter Verwendung eines CAS wie MuPAD realisierbar, welches die koordinatengeometrische Beschreibung einfacher Grundkörper zulässt.<sup>25</sup> Das folgende Beispiel zeigt, dass die Beschreibung aus Grundkörpern zusammengesetzter Objekte in MuPAD ähnlich wie in POV-Ray erfolgen kann.<sup>26</sup>

```

RumpfUnten := plot::Sphere(3.5, [0,-5.5,0],
                                Color=RGB::Gray):
RumpfOben := plot::Sphere(2.5, [0,0,0], Color=RGB::Gray):
Kopf := plot::Sphere(1.5, [0,3.5,0], Color=RGB::Gray):
Hut1 := plot::Cylinder(2, [0,4.7,0], [0,4.8,0],
                                Color=RGB::Black):
Hut2 := plot::Cylinder(0.9, [0,4.7,0], [0,6.8,0],
                                Color=RGB::Black):
Nase := plot::Cone(0.25, [0,3.5,-1.5], [0,3,-4.5],
                                Color=RGB::Red):
Auge1 := plot::Sphere(0.25, [-0.5,4,-1.32],
                                Color=RGB::Black):
Auge2 := plot::Sphere(0.25, [0.5,4,-1.32],
                                Color=RGB::Black):
Knopf1 := plot::Sphere(0.25, [0,1.5,-2], Color=RGB::Black):
Knopf2 := plot::Sphere(0.25, [0,0,-2.5], Color=RGB::Black):
Knopf3 := plot::Sphere(0.25, [0,-1.5,-2],
                                Color=RGB::Black):

plot(RumpfUnten,RumpfOben, Kopf, Nase,
     Hut1, Hut2, Auge1, Auge2, Knopf1, Knopf2, Knopf3)

```

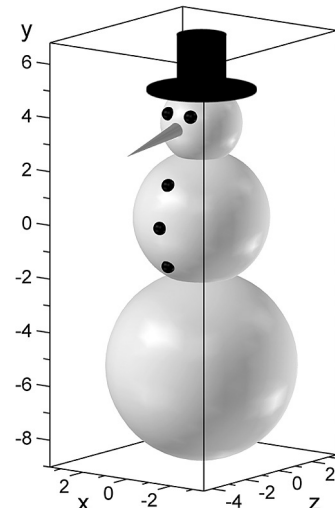


Abbildung 4.10: Mithilfe des CAS MuPAD generierter Schneemann

<sup>25</sup>Andere CAS können durch Makros bzw. Prozeduren, welche die Darstellung von Kugeln, Kegeln usw. durch Eingabe der Koordinaten bestimmender Punkte sowie Radien ermöglichen, erweitert werden. Die Anfertigung derartiger Makros durch die Schüler erscheint bei dem hier vorgeschlagenen Einstieg in das Stoffgebiet jedoch nicht sinnvoll, da diese hierzu zuerst Gleichungen für die darzustellenden Objekte aufstellen müssten und erst dann mit den koordinatengeometrischen Beschreibungen arbeiten könnten. Insofern erschien es sinnvoll, entsprechende Module zur Verfügung zu stellen.

<sup>26</sup>Im Interesse der Vergleichbarkeit mit dem von einer Schülerin in POV-Ray erstellten Beispiel (siehe S. 229) ist die MuPAD-Beschreibung unter Verwendung konkreter Zahlen angegeben. Natürlich ist jedoch auch in einem CAS die in Abschnitt 4.1.5 beschriebene Verwendung von Variablen sinnvoll und sollte im Verlauf des Unterrichts angestrebt werden.

### 4.1.8 Mögliche Wege der Weiterführung des Unterrichts

Nach dem in diesem Abschnitt beschriebenen Einstieg in das Stoffgebiet Analytische Geometrie besteht eine Reihe von Möglichkeiten der Vertiefung sowie der Fortführung des Stoffgebietes unter Anknüpfung an die hier beschriebenen Betrachtungen.

- Die *Modellierung* geometrischer und der Realität nachempfundener Objekte in einer Grafiksoftware kann unter Einbeziehung anspruchsvollerer, dabei dennoch durch elementare koordinatengeometrische Beschreibungen darzustellender Objekte wie *Boolescher Operationen*, *Prismen*, *Polyeder* und *Rotationskörper* fortgesetzt werden (siehe die Abschnitte 2.3.4 und 2.3.7 sowie die für Schüler verfasste Kurzanleitung auf S. 332).<sup>27</sup> Für die Anordnung derartiger Objekte im Raum sind im Gegensatz zur Positionierung der bislang betrachteten Körper, die durch die Koordinaten von Punkten erfolgt, elementare *Beschreibungen geometrischer Abbildungen* (Verschiebungen und Drehungen) notwendig, vgl. 4.4.3.<sup>28</sup>
- Erfahrungsgemäß besteht bei vielen Schülern ein großes Interesse an der *Erstellung von Animationen*; schon einfache Kameraflüge um einen zuvor modellierten Schneemann können als reizvoll empfunden werden. Während die technischen Aspekte der Generierung von Animationen bzw. Videos recht leicht zu bewältigen sind (siehe S. 339), erfordern anspruchsvollere Animationen die *Beschreibung von Bewegungskurven durch Parameter* oder die *zeitabhängige Darstellung geometrischer Transformationen*. Es ergeben sich somit Anknüpfungspunkte zur Behandlung von Parameterdarstellungen (siehe 4.3) sowie von Transformationen (4.4.4).
- In Ansätzen bereits thematisierte *Schnittbetrachtungen* (siehe S. 220) können aufgegriffen und weitergeführt werden; bei Verfolgung dieser Richtung bietet sich u. a. eine von raumgeometrischen Überlegungen ausgehende Behandlung der *Kegelschnitte* an (siehe 4.5.1) – als Vorbereitung hierfür wäre natürlich die auf S. 223 vorgeschlagene Herleitung der Kegelgleichung von besonderem Interesse.
- Ausgehend von der Herleitung von Kugel- und Kegelgleichungen erscheint eine Variation dieser Gleichungen für die *Beschreibung weiterer Flächen* sinnvoll (sie-

<sup>27</sup>Hierbei würden sich Bezüge zur Behandlung von Bézier- und Splinekurven anbieten (vgl. 2.3.7), falls diese z. B. Gegenstand eines vorherigen Stoffgebietes zur ebenen analytischen Geometrie gewesen wären. Obwohl dazu interessante Vorschläge vorliegen (vgl. 1.4.5), wird diese Voraussetzung nur in wenigen Fällen gegeben sein. Wie die Anleitung auf S. 332 verdeutlicht, ist die Behandlung von Rotationsflächen bzw. -körpern, die durch Splines generiert werden, jedoch auch ohne die ausführliche Thematisierung dieser Kurven möglich und m. E. sinnvoll. Bei Betrachtung von Funktionsgraphen als erzeugende Kurven von Rotationskörpern ergeben sich auch Anknüpfungspunkte an das Stoffgebiet Integralrechnung, in dem häufig Volumina von Rotationskörpern berechnet werden.

<sup>28</sup>Es wird im Folgenden auf die geometrische Modellierung von Objekten mithilfe von Booleschen Operationen, Prismen und Rotationskörpern nicht weiter eingegangen; entsprechende Herangehensweisen wurden bereits in den Abschnitten 2.3.4 und 2.3.7 sichtbar. Einige Ergebnisse von Schülern, die in einem Unterrichtsprojekt Objekte auf diese Weise modellierten, zeigen Abbildungen auf S. 361f. Weitere Arbeiten von Schülern und Studierenden, die in vielfältiger Weise die genannten Modellierungstechniken nutzten, befinden sich in den Ordnern *Schuelerarbeiten\Projekt-FPG-Cottbus-ma13-F-Rieper* und *Studierenden-Arbeiten* (siehe S. 365).

he hierzu 4.5.2 und 4.5.4), wozu insbesondere Flächen zweiter Ordnung zählen. Computervisualisierungen legen Umkehrüberlegungen zur Beschreibung anschaulich „vorgestellter“ Formen durch Gleichungen nahe, welche wiederum Schnittbetrachtungen mit Koordinaten- und dazu parallelen Ebenen erfordern.

- Experimente mit Farben und vor allem *Oberflächenstrukturen* (Texturen) sind erfahrungsgemäß für viele Schüler interessant; hierdurch ergeben sich nach der Erstellung erster Grafiken Ansatzpunkte, *Beleuchtungsmodelle* und (auf deren Grundlage) Parameter der Oberflächengestaltung zu thematisieren. Hierbei kann jedoch nicht unmittelbar an den hier beschriebenen Einstieg angeknüpft werden, vielmehr sind dazu Inhalte der Vektorgeometrie (Skalarprodukt, Normalenvektoren) erforderlich; in Abschnitt 4.6 wird darauf näher eingegangen.
- In vielen Fällen wird es aus Zeitgründen und wegen curricularer Vorgaben notwendig sein, anschließend an die hier geschilderten Überlegungen *traditionelle Standardinhalte des Stoffgebietes* zu behandeln. Ein gewisser „Bruch“ ist dabei unvermeidlich.<sup>29</sup> Dennoch sollte an den beschriebenen Einstieg zumindest insofern angeknüpft werden, als die durch die Schüler erlangten Fähigkeiten im Umgang mit der verwendeten Software im weiteren Verlauf des Unterrichts für Visualisierungen genutzt werden (siehe Abschnitt 4.2). Sinnvoll erscheint darüber hinaus, wie bereits in Abschnitt 3.3.1 ausgeführt wurde, ein Aufgreifen der in der Computergrafik (und allgemein der Informatik) gängigen Vektorauffassung bei der Behandlung von Vektoren.

Diese Aufzählung von Möglichkeiten, auf die im Folgenden noch näher eingegangen wird, deutet die Vielfalt und Reichhaltigkeit von Inhalten der analytischen Geometrie und der Computergrafik an, die in Anknüpfung an den beschriebenen Einstieg in das Stoffgebiet thematisiert werden können. Aus Zeitgründen wird es in keinem Falle möglich sein, alle genannten Bezugspunkte aufzugreifen, obwohl es m. E. jeder der genannten Gegenstände wert wäre, im Unterricht behandelt zu werden. Bei einer Durchführung projektorientierten Unterrichts besteht die Möglichkeit, Schülern bzw. Schülergruppen Entscheidungsmöglichkeiten einzuräumen, in welche Richtung sie ihre Arbeit weiterführen und welche Aspekte sie vertiefen. Ein entsprechendes Unterrichtsprojekt, bei dem sich die Schüler zwischen anspruchsvollen geometrischen Modellierungen, der Erstellung von Animationen sowie der Untersuchung von Kegelschnitten entscheiden konnten, führte F. RIEPER in einem Grundkurs am Fürst-Pückler-Gymnasiums in Cottbus durch, wobei die Schüler zum großen Teil anspruchsvolle Ergebnisse erreichten (siehe Abschnitt 5.3).

---

<sup>29</sup>Diese Situation war bei dem bereits mehrfach erwähnten Unterrichtsversuch gegeben, dessen nähere Beschreibung Gegenstand von Abschnitt 5.1 ist. Auf die Problematik des „Bruches“ zwischen dem Einstieg in das Stoffgebiet und der danach erfolgten Behandlung vektorieller Geraden- und Ebenengeometrie wird dabei einzugehen sein.

## 4.2 Visualisierung einiger Standardinhalte des Stoffgebietes Analytische Geometrie

### Einordnung und Ziele

In diesem Abschnitt werden Möglichkeiten beschrieben, eine Grafiksoftware bzw. ein CAS für die Visualisierung der die meisten Curricula der analytischen Geometrie bestimmenden Inhalte einer vektoriellen Raumgeometrie zu nutzen. Dabei erfolgt zunächst eine Beschränkung auf Objekte und Beziehungen der affinen Geometrie; das Skalarprodukt sowie metrische Beschreibungen geometrischer Objekte sind Gegenstände von Abschnitt 4.6.<sup>30</sup> Bei den im Folgenden dargestellten Beispielen zur Nutzung des Computers kommt ausschließlich der „Werkzeugaspekt“ (W, siehe S. 189) zum Tragen. Die Beispiele beziehen sich auf Themen und Aufgaben, die nach den meisten gegenwärtigen Rahmenplänen Unterrichtsgegenstände sind; ihr Aufgreifen erfordert somit keine konzeptionellen Veränderungen am gegenwärtigen Unterricht. Die hauptsächlichen *Ziele* der folgenden Vorschläge ergänzen somit lediglich die traditionellen Unterrichtsziele:

- Durch selbst erstellte grafische Darstellungen werden algebraische Beschreibungen geometrischer Objekte und rechnerische Untersuchungen von Lagebeziehungen um Veranschaulichungen ergänzt. Dadurch können Schüler mit oftmals nur rein kalkülmäßig gelösten Standardaufgaben anschauliche Vorstellungen verbinden und sie erhalten ein Werkzeug, um ihre Ergebnisse visuell zu überprüfen.
- Die Schüler entdecken durch visuelle Experimente einige Eigenschaften und Beziehungen, deren Erarbeitung ansonsten recht schwierig ist und die deshalb oft nur mitgeteilt werden, selbst.

Diese didaktischen Intentionen könnten zum großen Teil auch unter Verwendung herkömmlicher räumlicher Modelle realisiert werden. Derartige Modelle sind jedoch häufig nicht in ausreichendem Maße vorhanden und ermöglichen aus mechanischen Gründen nicht alle der hier dargestellten Betrachtungen. Dennoch sollten herkömmliche Veranschaulichungsmittel zum Einsatz kommen, wann immer dies möglich ist, so dass dabei auch der Bezug zu Computerdarstellungen deutlich wird und die Schüler damit klarere Vorstellungen der abgebildeten räumlichen Sachverhalte gewinnen.

Die hier beschriebene Nutzung des Computers zu Visualisierungszwecken erfolgt zum größten Teil im *Zusammenhang mit Aufgaben*, welche die Schüler selbstständig bearbeiten können.<sup>31</sup> Als *Software* kann dazu *POV-Ray*, ergänzt um die in Anhang A auf S. 335f. und S. 343ff. beschriebenen Erweiterungen *anageoL* bzw. *anageoR*, zum Einsatz kommen. Auch die Nutzung eines CAS oder einer speziellen Software wie *DreiDGeo*

<sup>30</sup>Mit dieser Gliederung soll nicht für eine Trennung affiner und metrischer Aspekte im Unterricht plädiert werden, vgl. die Bemerkungen auf S. 227f. Sinnvoll ist u. a. eine „Verzahnung“ der in diesem Abschnitt beschriebenen Überlegungen mit der Behandlung des Skalarproduktes (siehe 4.6).

<sup>31</sup>In den folgenden Ausführungen werden hauptsächlich entsprechende Aufgaben und zugehörige Lösungen dargestellt. Da sich die Aufgaben auf bekannte Standardinhalte des Stoffgebietes beziehen, sind umfangreichere Erläuterungen dazu in den meisten Fällen nicht notwendig.

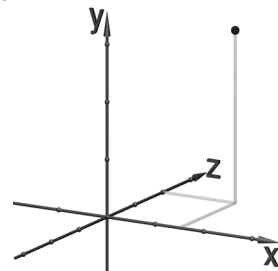
(siehe 2.7.6 und 3.4) ist möglich. In diesem Abschnitt werden Lösungen unter Verwendung von POV-Ray und (anhand einiger Beispiele) des CAS *MuPAD* angegeben.

Wie bereits ausgeführt wurde, liegt POV-Ray ein *linkshändiges Koordinatensystem* zugrunde. Für die im Folgenden verwendeten Befehle lässt sich jedoch auch mit POV-Ray eine Darstellung in einem rechtshändigen KS erreichen. Dies kann sinnvoll sein, wenn direkte Vergleiche mit Illustrationen in Schulbüchern vorgenommen werden sollen.<sup>32</sup> Einige Abbildungen in diesem Abschnitt enthalten sowohl mit POV-Ray erzeugte Versionen in einem links- als auch MuPAD-Versionen bezüglich eines rechtshändigen KS.

### 4.2.1 Veranschaulichung der Lage von Punkten im Koordinatensystem sowie von Vektoren durch Pfeile

Punkte werden in einem räumlichen Koordinatensystem i. Allg. durch kleine Kugeln dargestellt, unter Verwendung der schon erwähnten **anageo**-Erweiterungen für POV-Ray ist dies durch Eingabe von **punkt** ( $\langle x, y, z \rangle$ , *textur*) möglich.<sup>33</sup>

Da die Lage eines Punktes bei der Betrachtung eines zweidimensionalen Bildes dabei nicht eindeutig erkennbar ist, enthalten die **anageo**-Erweiterungen zusätzlich Makros für die Darstellung eines Punktes mit dem Lot auf die  $x$ - $z$ - bzw. die  $x$ - $y$ -Ebene und den Loten von dessen Fußpunkt auf die entsprechenden Koordinatenachsen (siehe die rechte Abb.); diese erweiterte Darstellung ist durch **pluspunkt** ( $\langle x, y, z \rangle$ , *textur*) aufzurufen.



Der Veranschaulichung von Vektoren durch Pfeile liegt die auf S. 200 dargestellte Einführung des Vektorbegriffs zugrunde. Wie dort bereits erwähnt wurde, lässt sich die Unterscheidung zwischen Ortsvektoren und „freien“ Vektoren (die z. B. als Richtungsvektoren auftreten) in vielen Zusammenhängen kaum vermeiden.<sup>34</sup> Bei der Darstellung durch Pfeile erscheint es sogar sinnvoll, drei Kontexte zu unterscheiden, in denen Vektoren in der analytischen Geometrie auftreten. Die **anageo**-Erweiterungen für POV-Ray enthalten daher Makros für die Pfeildarstellung von Ortsvektoren, von Verbindungsvektoren zweier Punkte sowie für die Antragung von Pfeilen an gegebene Punkte:<sup>35</sup>

<sup>32</sup>Für beide Versionen stehen Vorlagen und Makros zur Verfügung, deren Nutzung sowie Vor- und Nachteile auf S. 335 kurz beschrieben sind. Die Abbildungen in diesem Abschnitt wurden unter Verwendung der Vorlage generiert, welche Objekte anhand des (linkshändigen) POV-Ray-Koordinatensystems darstellt. Die entsprechenden POV-Ray-Dateien befinden sich in dem Ordner 4-2 (siehe Anhang D, S. 365). Durch Verändern der Zeile `#include "anageoL.inc"` zu `#include "anageoR.inc"` in den Köpfen dieser Dateien werden grafische Darstellungen bezüglich eines rechtshändigen KS erzeugt.

<sup>33</sup>Für **textur** genügt es, ein Schlüsselwort anzugeben (siehe S. 335f.).

<sup>34</sup>Wenngleich es diese Unterscheidung in Bezug auf Vektoren (als mathematische Objekte) selbst nicht gibt und sie lediglich aus der Art ihrer Verwendung resultiert, tritt sie bei der Darstellung von Vektoren als Pfeile z. B. im Zusammenhang mit Parameterdarstellungen unweigerlich in Erscheinung.

<sup>35</sup>Natürlich würde eine einzige dieser Anweisungen ausreichen, um Pfeildarstellungen von Vektoren in allen drei Kontexten zu erstellen: Ortsvektoren lassen sich als Verbindungsvektoren auffassen und die Antragung des einen Vektor  $\vec{v}$  repräsentierenden Pfeils an einen Punkt  $P$  wäre auch als Verbindungsvektor der Punkte  $P$  und  $P + \vec{v}$  möglich sowie umgekehrt. Dennoch ist es m. E. sinnvoll, für den



- Ortsvektor eines Punktes  $P(x; y; z)$ :  
`ortsvektor (<x,y,z>, textur)`
- Verbindungsvektor zweier Punkte  $P_1(x_1; y_1; z_1)$  und  $P_2(x_2; y_2; z_2)$ :  
`verbindungsvektor (<x1,y1,z1>, <x2,y2,z2>, textur)`
- Pfeildarstellung eines Vektors  $\begin{pmatrix} x \\ y \\ z \end{pmatrix}$ , angetragen an einen Punkt  $P(x_P, y_P, z_P)$ :  
`vektoranpunkt (<xP,yP,zP>, <x,y,z>, textur)`

Nach der Einführung von Vektoren können die Schüler einige einfache Aufgaben bearbeiten, bei denen sie diese Anweisungen nutzen.<sup>36</sup> Anhand der folgenden Aufgabe lässt sich u. a. herausarbeiten, dass verschiedene Pfeile denselben Vektor repräsentieren.

- A1. Öffnen Sie die Vorlagedatei **anageoL.pov** und speichern Sie diese unter dem Namen **Aufgabe1.pov**. Stellen Sie 4 Punkte (als **pluspunkt**) im sichtbaren Bereich dar (Koordinaten von  $-5 \dots 5$ ). Blenden Sie das Koordinatensystem mit ein. Wählen Sie einen Vektor mit Koordinaten zwischen  $-3$  und  $3$ . Tragen Sie an jeden der von Ihnen dargestellten Punkte einen Pfeil an, der den von Ihnen gewählten Vektor darstellt (Befehl **vektoranpunkt**).

Für die Anfertigung von Grafiken zu Aufgabe A1 (siehe Abb. 4.11) sind in folgender Form Anweisungen in POV-Ray (links) bzw. MuPAD (rechts) einzugeben:

<code>ks</code>	<code>a:=matrix([-2,3,-1])</code>
<code>#declare a=&lt;-2,3,-1&gt;;</code>	<code>P:=matrix([4,1,5])</code>
<code>#declare P=&lt;4,1,5&gt;;</code>	<code>PunktP:=plot::Point3d(P)</code>
<code>pluspunkt(P, rot_matt)</code>	<code>PfeilAnP:=plot::Arrow3d(P,P+a)</code>
<code>vektoranpunkt(P, a, silbergrau)</code>	<code>plot(PunktP, PfeilAnP)</code>

Die Angabe der Koordinaten mittels der **matrix**-Anweisung ist in MuPAD für die Darstellung von Pfeilen notwendig (vgl. S. 195), Punkte können auch mittels direkter

Einstieg die aufgeführten drei kontextabhängigen Anweisungen zur Verfügung zu stellen. Nach einiger Zeit lässt sich dann herausarbeiten, dass Verbindungsvektoren mithilfe von „**vektoranpunkt**“ sowie Pfeile an gegebene Punkte durch „**verbindungsvektor**“ darstellbar sind (vgl. Aufgabe A3 auf S. 236).

<sup>36</sup>Die hier vorgestellten Aufgaben stehen exemplarisch für eine Reihe ähnlicher Aufgaben, welche Schüler (auch als Hausaufgaben) bearbeiten können. Lösungsdateien für alle Aufgaben befinden sich in dem Ordner 4-2; die Dateinamen entsprechen der Nummerierung der Aufgaben.

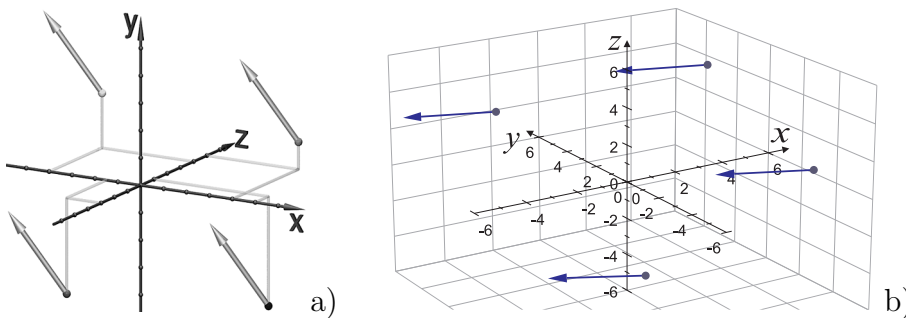


Abbildung 4.11:  
 Lösungsbeispiele  
 für Aufgabe A1:  
 a) mit POV-Ray,  
 b) mit MuPAD

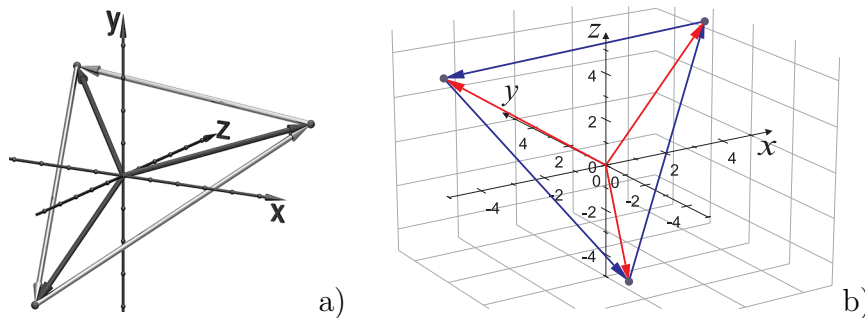


Abbildung 4.12:  
Lösungsbeispiele  
für Aufgabe A2:  
a) mit POV-Ray,  
b) mit MuPAD

Eingabe `plot::Point3d(x,y,z)` gezeichnet werden. Wie bereits in Abschnitt 4.1.5 ausgeführt wurde, ist es sinnvoll, die Schüler auf die Vorteile von Definitionen der Form `#declare` bzw. „:=“ für Objekte hinzuweisen, die mehrfach genutzt werden. So wird in diesem und in den folgenden Beispielen auf Koordinatentripel mehrfach zugegriffen.

- A2. Löschen Sie in Ihrer Arbeitsdatei aus Aufgabe A1 einen Punkt, so dass die drei am besten sichtbaren Punkte  $P$ ,  $Q$  und  $R$  übrig bleiben. Stellen Sie die Ortsvektoren dieser 3 Punkte dar. Welche Koordinaten haben die Ortsvektoren?

Stellen Sie die drei Verbindungsvektoren  $\overrightarrow{PQ}$ ,  $\overrightarrow{QR}$  und  $\overrightarrow{RP}$  dar.<sup>37</sup>

- A3. Berechnen Sie die Koordinaten der von Ihnen in Aufgabe A2 dargestellten Verbindungsvektoren und stellen Sie diese nun mit dem Befehl `vektoranpunkt` dar. (Löschen Sie zunächst die zuvor mittels `verbindungsvektor` erzeugten Pfeile.)<sup>38</sup>

Die folgende Aufgabe soll vor allem einen Beitrag dazu leisten, dass den Schülern die „Isomorphie“ des  $n$ -Tupel- und des Pfeilklassenmodells hinsichtlich der Addition von Vektoren deutlich wird (vgl. S. 199).<sup>39</sup>

- A4. Gegeben sind die Vektoren

$$\vec{a} = \begin{pmatrix} 5 \\ 4 \\ -2, 5 \end{pmatrix} \text{ und } \vec{b} = \begin{pmatrix} -4 \\ -1 \\ -2 \end{pmatrix}.$$

Berechnen Sie die Summe dieser beiden Vektoren und stellen Sie  $\vec{a}$ ,  $\vec{b}$  und  $\vec{a} + \vec{b}$  als Pfeile so dar, dass die  $\vec{a}$  und  $\vec{a} + \vec{b}$  kennzeichnenden Pfeile im Koordinatenursprung beginnen und der  $\vec{b}$  beschreibende Pfeil von der Spitze des zu  $\vec{a}$  gehörenden Pfeils ausgeht.

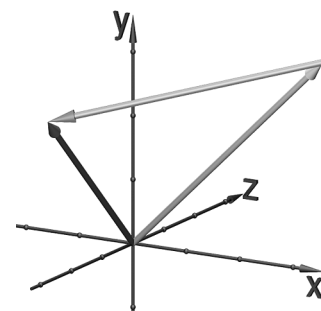


Abbildung 4.13: Aufgabe A4

<sup>37</sup>Um dieser Aufgabe entsprechende Grafiken zu erstellen (Abb. 4.12), sind in POV-Ray die auf S. 235 erklärten Makros `ortsvektor` und `verbindungsvektor` zu nutzen; in MUPAD lassen sich Verbindungsvektoren durch `Arrow3d(P,Q)` und als Spezialfall Ortsvektoren durch `Arrow3d(P)` erzeugen.

<sup>38</sup>Es ergeben sich dieselben grafischen Darstellungen wie bei Aufgabe A2 (siehe Abb. 4.12).

<sup>39</sup>Es ist damit ein elementares Verständnis der Isomorphie beider Modelle gemeint, wobei der Begriff „Isomorphie“ nicht verwendet und thematisiert werden muss, siehe S. 199. Analoge Aufgaben können auch bezüglich der Multiplikation von Vektoren mit Skalaren gestellt werden.

### 4.2.2 Veranschaulichung von Geraden im Raum und Lagebeziehungen zwischen ihnen

In Abschnitt 3.3.2 wurde darauf eingegangen, dass es für die Herausbildung eines Verständnisses von Geraden als Punktmengen sinnvoll ist, wenn die Schüler Geraden aus einzelnen Punkten „aufbauen“. Die folgende Aufgabe kann unmittelbar nach der Einführung von Vektoren und einigen Aufgaben wie A1–A4 gestellt werden und die Einführung von Parameterdarstellungen vorbereiten.

- A5. Gegeben sind der Punkt  $P(0,5; 1; 1,5)$  und der Vektor  $\vec{a} = \begin{pmatrix} -2,5 \\ 1 \\ -1,5 \end{pmatrix}$ . Stellen Sie  $P$  als Punkt und  $\vec{a}$  als in  $P$  beginnenden Pfeil dar. Fügen Sie Ihrer Darstellung die Punkte  $P - 2 \cdot \vec{a}$ ,  $P - 1,5 \cdot \vec{a}$ ,  $P - \vec{a}$ ,  $P - 0,5 \cdot \vec{a}$ ,  $P + 0,5 \cdot \vec{a}$ ,  $P + \vec{a}$ ,  $P + 1,5 \cdot \vec{a}$  und  $P + 2 \cdot \vec{a}$  hinzu. Betrachten Sie die Darstellung aus verschiedenen Richtungen.

Ergebnisse der Bearbeitung dieser Aufgabe mithilfe von POV-Ray bzw. MuPAD zeigen Abb. 4.14 a) sowie Abb. 4.15 a) auf S. 238.<sup>40</sup> Daran wird spätestens nach Betrachtung aus verschiedenen Richtungen deutlich, dass alle (durch kleine Kugeln dargestellten) „Punkte“ auf einer Geraden liegen. Nach der Darstellung einer größeren Zahl von Punkten durch die Verkleinerung der Parameterabstände (Abb. 4.14 b) dürfte den Schülern auch (zumindest anschaulich) plausibel werden, dass sich umgekehrt alle Punkte der durch  $P$  verlaufenden Geraden, deren Richtung durch  $\vec{a}$  gegeben ist, in der Form  $P + t \cdot \vec{a}$  mit  $t \in \mathbb{R}$  darstellen lassen.<sup>41</sup>

Steht Zeit zur Verfügung, um einfache *Schleifen* oder *Prozeduren* zu behandeln, so kann durch die Darstellung sehr viele Punkte, die allmählich zu einer Strecke „verschmelzen“, noch anschaulicher werden, dass die Punktmenge  $\{X = P + t\vec{a} \mid t \in \mathbb{R}\}$  eine Gerade ist. So besteht die Strecke in Abb. 4.14 c) aus 400 kleinen Kugeln und wurde in POV-Ray durch folgende **while**-Schleife erzeugt:

<sup>40</sup>Um Darstellungen wie in Abb. 4.14 zu erstellen, lassen sich die auf S. 234f. beschriebenen Anweisungen **punkt** bzw. **pluspunkt** sowie **vektoranpunkt** nutzen. Bei der Lösung mithilfe von MuPAD kommen die ebenfalls bereits genannten Grafikobjekte **Point3d** und **Arrow3d** zum Einsatz; nach Festlegung von  $P$  und  $\vec{a}$  lassen sich die einzelnen Punkte jeweils durch **plot::Point3d(P-2\*a)** usw. generieren.

<sup>41</sup>Nicht berücksichtigt wird bei den beschriebenen Überlegungen die unendliche Ausdehnung von Geraden; natürlich können nur endliche Parameterwerte betrachtet werden. Dieses Problem gilt für jede grafische Darstellung unabhängig von den verwendeten Hilfsmitteln. Die Unendlichkeit der Ausdehnung kann den Schülern nur mitgeteilt werden, während grafische Darstellungen immer nur Strecken zeigen. Allerdings wird diese Frage bereits in der SI, ansatzweise sogar in der Grundschule thematisiert.

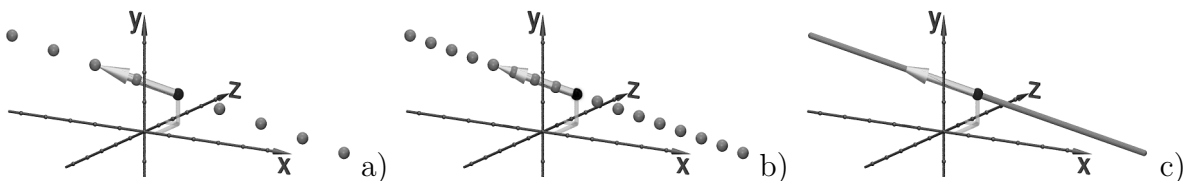


Abbildung 4.14: Aufbau einer durch eine Parameterdarstellung gegebenen Geraden

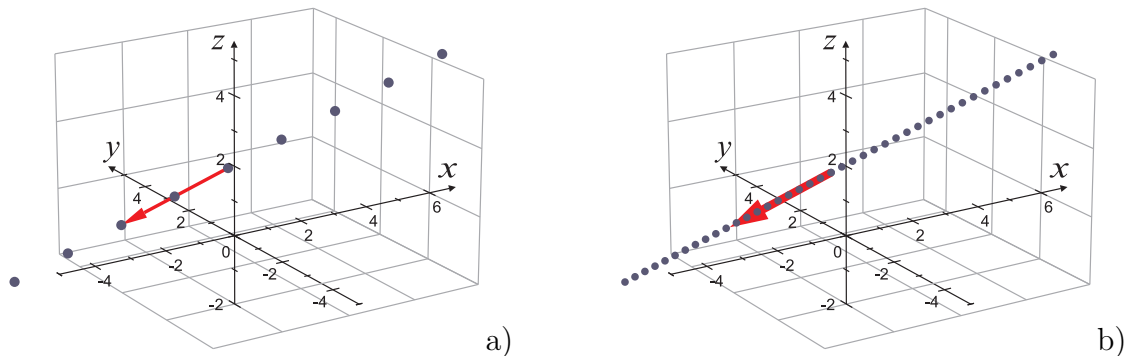


Abbildung 4.15: a) Lösungsgrafik zu Aufgabe A5; b) Erzeugung einer großen Zahl von Punkten mithilfe einer MuPAD-Prozedur

```
#declare i=-200;
  #while (i <= 200)
    punkt(P+i*a/100 blau_matt)
  #declare i=i+1;
#end
```

Bei Verwendung von MuPAD besteht der einfachste Weg, sehr viele Punkte einer Geraden darzustellen, in der Erstellung einer einfachen Prozedur:

```
Puenktchen := proc(i)
begin
  plot::Point3d(P+i/10*a);
end_proc:
plot(Puenktchen(i) $ i = -20..20)
```

Mithilfe dieser Prozedur wurde Abb. 4.15 b) erstellt; die Erzeugung einer größeren Zahl von Kugeln auf derselben Strecke ist möglich, indem z. B. `plot::Point3d(P+i/100*a)` und in der `plot()`-Anweisung `$ i = -200..200` gesetzt wird. Durch den allmählichen Übergang von einzelnen zu sehr vielen „Punkten“, die als Strecke wahrgenommen werden, können die Schüler den Charakter einer Strecke bzw. einer Geraden als Punktmenge plastisch und konstruktiv nachvollziehen.<sup>42</sup>

Wie bereits in Abschnitt 3.3.2 ausgeführt wurde, dürfte es hinsichtlich der Ausprägung des Punktmengengedankens und der Erfassung funktionaler Aspekte im Zusammenhang mit Parameterdarstellungen nicht ausreichend sein, lediglich Geraden zu betrachten. Die beschriebenen Vorgehensweisen lassen sich jedoch verallgemeinern und zu Abb. 4.14 bzw. 4.15 analoge Darstellungen für beliebige durch Parameterdarstellungen gegebene Kurven erstellen; nähere Ausführungen dazu enthält Abschnitt 4.3.

### Lagebeziehungen zwischen Geraden im Raum

Für die Untersuchung von Lagebeziehungen ist es notwendig, Geraden durch die Angabe der Koordinaten zweier Punkte bzw. eines Punktes und eines Richtungsvektors darzustellen. Dazu steht in den `anageo`-Vorlagen für POV-Ray der Befehl

<sup>42</sup>Ein Video, bei dem eine Gerade allmählich durch eine große Zahl von Punkten aufgebaut wird, enthält der Ordner 4-2\Videos-Punktmengen.

`gerade (<x1,y1,z1>, <x2,y2,z2>, textur)`

zur Verfügung, der die durch die Punkte  $P_1(x_1; y_1; z_1)$  und  $P_2(x_2; y_2; z_2)$  verlaufende Gerade darstellt. In MuPAD ist mittels des Befehls `plot::Line3d` auf analoge Weise nur die Darstellung von Strecken mit gegebenen Endpunkten möglich. Für Lagebetrachtungen ist es jedoch i. Allg. sinnvoll, weiter reichende Ausschnitte von Geraden darzustellen. Dazu kann der Befehl für die Darstellung von Raumkurven genutzt werden. Die Gerade durch den Punkt  $P_0$  mit dem Richtungsvektor  $\vec{a}$  lässt sich innerhalb des Intervalls  $[2; 2]$  für den Parameter  $t$  durch

`g:=plot::Curve3d(P0+t*a, t=-2..2)`

mithilfe von MuPAD darstellen. Die Koordinatentripel von  $P_0$  und  $\vec{a}$  sind dazu in der auf S. 235 angegebenen Weise mittels der `matrix`-Anweisung einzugeben.

Mit der folgenden Aufgabe A6 können Schüler lernen, die Anweisung für die Darstellung von Geraden zu verwenden, wobei sie zunächst die Koordinaten eines zweiten Punktes berechnen müssen. Weiterhin ist beabsichtigt, dass sie rechnerische und grafische Vorgehensweisen bei der Entscheidung, ob gegebene Punkte zu einer Geraden gehören, kombinieren. Durch die Bearbeitung einiger Aufgaben dieser Art (zu denen auch A7 gehört) wird für die Schüler deutlich, wie sie die Software auch für die grafische Ergänzung und Kontrolle rechnerisch zu lösender Aufgaben (z. B. typischer Schulbuchaufgaben) nutzen können.

A6. Gegeben ist eine Gerade  $g$  durch eine Parametergleichung:

$$g: P = \begin{pmatrix} 2 \\ 4 \\ 3 \end{pmatrix} + r \cdot \begin{pmatrix} 4 \\ -6 \\ -5 \end{pmatrix} \quad (r \in \mathbb{R}).^{43}$$

- Stellen Sie die Gerade  $g$  sowie ihren Aufpunkt und den Richtungsvektor dar. (Sie benötigen dazu 2 Punkte der Geraden. Wie erhalten Sie die Koordinaten eines zweiten Punktes?)
- Stellen Sie die Punkte  $A(7; 3; 8)$ ,  $B(4; -7; -6)$  und  $C(0; 7; 5,5)$  in derselben Datei wie die Gerade  $g$  dar.
- Vergrößern Sie den Wert für `intervall` so weit, bis Sie alle drei Punkte sehen. Blenden Sie das Koordinatensystem ein. Betrachten Sie die Szene aus verschiedenen Blickwinkeln. Welche der Punkte gehören zu  $g$ ?
- Begründen Sie Ihre Ergebnisse auch rechnerisch.

---

<sup>43</sup>Die für die Parameterdarstellung verwendete Schreibweise ist etwas ungewöhnlich; in den meisten Schulbüchern treten Bezeichnungen wie  $g: \vec{x} = \begin{pmatrix} 2 \\ 4 \\ 3 \end{pmatrix} + r \cdot \begin{pmatrix} 4 \\ -6 \\ -5 \end{pmatrix}$  oder  $g: P = (2; 4; 3) + r \cdot \begin{pmatrix} 4 \\ -6 \\ -5 \end{pmatrix}$  auf. Beide Varianten haben jedoch Nachteile: die Ortsvektorschreibweise ( $\vec{x}$  für  $P$ ) erscheint Schülern oft gekünstelt, da ja *Punkte* einer Geraden beschrieben werden; die Kombination von Zeilen- und Spaltenschreibweise ist hingegen unübersichtlicher in Bezug auf die Zuordnung der einander entsprechenden Koordinaten. Die in Aufgabe A6 angegebene Schreibweise ist gerechtfertigt, wenn sowohl Punkte als auch Vektoren als Tripel aufgefasst werden (siehe [36], S. 175f.) bzw. die Parametergleichung als Gleichung für Koordinatentripel interpretiert wird.

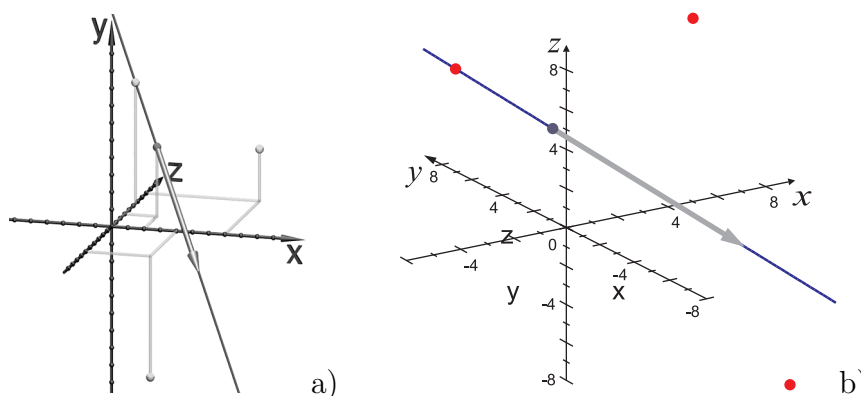


Abbildung 4.16:  
Lösungsgrafiken  
zu Aufgabe A6:  
a) in POV-Ray,  
b) in MuPAD

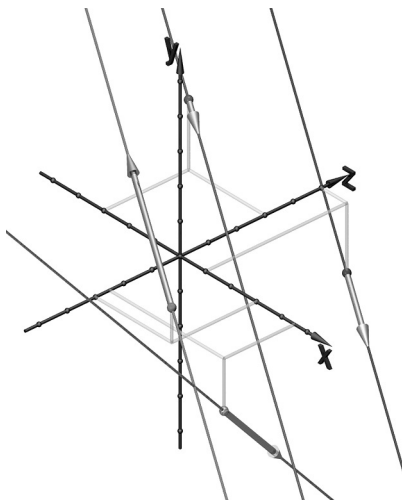
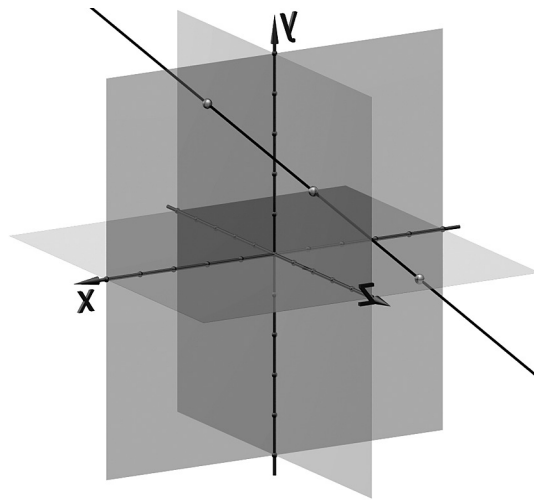
Grafiken, die bei der Lösung dieser Aufgabe anzufertigen sind, zeigt Abb. 4.16.<sup>44</sup> Die rechnerische Ermittlung der Zugehörigkeit eines Punktes zu einer Geraden muss vorher im Unterricht nicht explizit (im Sinne der Erarbeitung eines Lösungsverfahrens) thematisiert werden, es ist dann jedoch großer Wert auf die Auswertung der Aufgabe zu legen. Erfahrungsgemäß gelangen einige Schüler (auch in Grundkursen) zu der Erkenntnis, dass sie das Koordinatentripel des betreffenden Punktes mit der rechten Seite der Parametergleichung gleichsetzen und überprüfen müssen, ob sich für alle drei Koordinatengleichungen derselbe Wert für den Parameter ergibt. Vielen Schülern gelingt dies jedoch nicht selbstständig. Entscheidend für den Erfolg dürfte sein, wie gründlich die Schüler zuvor den Charakter von Geraden als Punktmengen erfasst haben.

Die folgende Aufgabe eignet sich als Einstieg in die Untersuchung von Lagebeziehungen zwischen Geraden.

A7. Gegeben sind vier Geraden  $g_1: P = \begin{pmatrix} 3 \\ 1 \\ -3 \end{pmatrix} + r \cdot \begin{pmatrix} -4 \\ 2 \\ 2 \end{pmatrix}$ ,  $g_2: P = \begin{pmatrix} 1 \\ -2 \\ 5 \end{pmatrix} + s \cdot \begin{pmatrix} 2 \\ -1 \\ -1 \end{pmatrix}$ ,  
 $g_3: P = \begin{pmatrix} -3 \\ 2 \\ 3 \end{pmatrix} + t \cdot \begin{pmatrix} 1 \\ -0,5 \\ -0,5 \end{pmatrix}$  und  $g_4: P = \begin{pmatrix} 4,5 \\ -1,5 \\ -2,5 \end{pmatrix} + u \cdot \begin{pmatrix} -1 \\ -3 \\ 3 \end{pmatrix}$  (mit  $r, s, t, u \in \mathbb{R}$ ).

- Stellen Sie diese vier Geraden in unterschiedlichen Farben sowie ihre Aufpunkte und Richtungsvektoren dar (notieren Sie die Farben der Geraden).
- Betrachten Sie die Geraden unter verschiedenen Winkeln und stellen Sie jeweils fest, ob zwei Geraden parallel oder windschief sind oder sich schneiden.
- Schätzen Sie die Koordinaten des Schnittpunktes (bzw. der Schnittpunkte) sich schneidender Geraden so genau wie möglich ab.
- \* Überlegen Sie, wie sich die Schnittpunktkoordinaten berechnen lassen.
- Formulieren Sie eine Aussage der Art: „Zwei Geraden sind parallel, wenn für Ihre Richtungsvektoren Folgendes gilt: ...“

<sup>44</sup>Die Lage der Geraden im Raum und die Positionen der Punkte sind aus diesen Abbildungen nicht eindeutig zu erkennen. Dazu ist es bei Verwendung von POV-Ray nötig, Ansichten aus unterschiedlichen Kamerapositionen bzw. mit verschiedenen Werten für **winkel** und **hoehe** zu generieren (siehe S. 213 bzw. die Anleitung auf S. 331). In MuPAD können Darstellungen interaktiv gedreht werden.

Abbildung 4.17:  
Grafik zu Auf-  
gabe A7Abbildung 4.18:  
Grafik zu Auf-  
gabe A8 (Spür-  
punkte)

Die Schüler können anhand von Aufgabe A7 herausfinden, dass zwischen Geraden Parallelität besteht, wenn ihre Richtungsvektoren jeweils Vielfache voneinander sind.<sup>45</sup> Eine mit POV-Ray erstellte Grafik zur Lösung der Aufgabe A7 zeigt Abb. 4.17. Um die Koordinaten des Schnittpunktes der Geraden  $g_1$  und  $g_4$  näherungsweise abzulesen und festzustellen, dass keine weiteren Schnittpunkte existieren, ist die Betrachtung mehrerer Ansichten erforderlich.

Häufig werden bei der Untersuchung der Lage von Geraden *Spurpunkte* (Schnittpunkte mit den Koordinatenebenen) berechnet. Die Behandlung von Ebenengleichungen ist dazu noch nicht erforderlich, da leicht überlegt werden kann, dass für Punkte auf Koordinatenebenen jeweils eine Koordinate Null sein muss. Die Berechnung der Spurpunkte ist vor allem notwendig, um den Verlauf von Geraden in manuell angefertigten Schrägbildern darzustellen. In Computergrafiken erleichtern transparente Koordinatenebenen oft die Orientierung. Bevor die Schüler Ebenengleichungen behandeln, können sie die Koordinatenebenen durch die Vorgabe entsprechender Anweisungen bzw. die Einbindung einer Datei, welche diese enthält, darstellen.<sup>46</sup> Die folgende Aufgabe kombiniert die Berechnung von Spurpunkten mit der grafischen Überprüfung der Ergebnisse.

A8. Berechnen Sie die Koordinaten der Spurpunkte der Geraden

$$g: P = \begin{pmatrix} -2 \\ 0 \\ 4 \end{pmatrix} + r \cdot \begin{pmatrix} 2 \\ 2 \\ -2 \end{pmatrix} \quad (r \in \mathbb{R}).$$

Stellen Sie die Gerade  $g$  sowie die Spurpunkte grafisch dar. Blenden Sie die Koordinatenebenen ein und überprüfen Sie die Ergebnisse Ihrer Berechnungen.

<sup>45</sup>In Unterrichtsversuchen gelangte die Mehrheit der Schüler zu dieser Erkenntnis, ohne dass darauf vorher eingegangen wurde. Ein Verfahren für die Berechnung des Schnittpunktes (Aufgabenteil \*) konnten die meisten Schüler eines Grundkurses jedoch nicht selbstständig entwickeln. Lediglich recht vage formulierte Ansätze wurden von einigen Schülern bei der Besprechung der Aufgabe vorgetragen.

<sup>46</sup>Eine geeignete Datei `koordinatenebenen.inc` befindet sich in dem Ordner 4-2 (siehe S. 365f.). Sie kann mittels `#include "koordinatenebenen.inc"` in jeder POV-Ray-Datei, in der zuvor eine der Erweiterungen `anageoL.inc` oder `anageoR.inc` eingebunden wurde, genutzt werden.

### 4.2.3 Darstellung von Ebenen; Schnittpunkte von Geraden und Ebenen

Zur Einführung in die Behandlung von Ebenen und ihrer Beschreibung durch Parameterdarstellungen bietet es sich (wie für Geraden, siehe S. 237f.) wiederum an, für einige Parameterpaare  $(s; t)$  Punkte der Form  $P = P_0 + s \cdot \vec{a} + t \cdot \vec{b}$  darzustellen, um den Punktmengencharakter von Ebenen für die Schüler deutlich werden zu lassen. Im Vergleich zu Geraden ist dazu jedoch eine größere Zahl von Punkten erforderlich, so dass die Verwendung von Schleifen, die im Falle von Ebenen verschachtelt werden müssen, besonders sinnvoll ist. Sind dafür die Voraussetzungen nicht gegeben und steht nicht genügend Zeit zur Verfügung, so können die Schüler zumindest einige wenige Punkte darstellen bzw. mittels Kopieren–Einfügen und die Veränderung von Parameterwerten deren Zahl erhöhen.

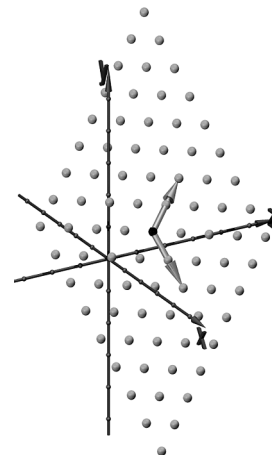


Abbildung 4.19:  
Punkte einer Ebene

Für die direkte grafische Darstellung von Ebenen mithilfe von POV-Ray stehen in den anageo-Erweiterungen folgende Anweisungen zur Verfügung:<sup>47</sup>

`ebenepar (<xP,yP,zP>, <xa,ya,za>, <xb,yb,zb>, textur)` – Ebene durch den Punkt  $P(x_P; y_P; z_P)$  mit den Richtungsvektoren  $\vec{a} = \begin{pmatrix} x_a \\ y_a \\ z_a \end{pmatrix}$  und  $\vec{b} = \begin{pmatrix} x_b \\ y_b \\ z_b \end{pmatrix}$ ;  
`ebene (A, B, C, D, textur)` – Ebene mit der Gleichung  $Ax + By + Cz + D = 0$ .

In MuPAD können für die Darstellung von Ebenen die für Flächen vorgesehenen Anweisungen genutzt werden (vgl. 2.3.5).<sup>48</sup> Mittels

`plot::Surface(P0 + s*a + t*b, s=smin..smax, t=smin..tmax)`<sup>49</sup> bzw.

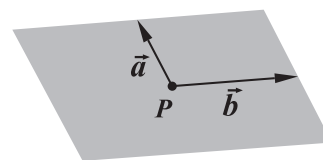
`plot::Implicit3d(A*x+B*y+C*z+D, x=xmin..xmax, y=ymin..ymax, z=zmin..zmax)`

lassen sich die Ebenen mit der Parameterdarstellung  $P = P_0 + s \cdot \vec{a} + t \cdot \vec{b}$  bzw. mit der impliziten Gleichung  $Ax + By + Cz + D = 0$  generieren. Die Angabe der Parameter- bzw. Koordinatenintervalle, innerhalb derer Darstellungen erfolgen, ist erforderlich.

Alternativ ist die Darstellung durch Parameterdarstellungen gegebener Ebenen mithilfe der MuPAD-Funktion

`plot::Parallelogram3d(P,a,b)`

möglich, die ein Parallelogramm mit dem Zentrum  $P$ , das von den Vektoren  $\vec{a}$  und  $\vec{b}$  aufgespannt wird, generiert.



<sup>47</sup>Bei Nutzung dieser Anweisungen werden dünne Quader erzeugt, welche die Lage der entsprechenden Ebenen veranschaulichen (siehe S. 335f. und S. 343ff.).

<sup>48</sup>Da die Berechnungsgeschwindigkeit von MuPAD wesentlich höher als die von POV-Ray ist, entstehen dadurch keine längeren Wartezeiten, die sich in POV-Ray bei Nutzung der mit `Implicit3d` bzw. `Surface` vergleichbaren `isosurface`- und `parametric`-Anweisungen unweigerlich ergeben.

<sup>49</sup> $P_0$ ,  $\vec{a}$  und  $\vec{b}$  sind dazu mittels der `matrix`-Anweisung anzugeben (siehe S. 235).



Soll ein größerer Ausschnitt der entsprechenden Ebene dargestellt werden, so sind Vielfache ihrer Richtungsvektoren einzusetzen.<sup>50</sup>

Anhand der folgenden Aufgaben können Schüler die Darstellung von Ebenen in der verwendeten Software erproben (wobei ihnen eine kurze Anleitung wie auf S. 335f. zur Verfügung stehen sollte) sowie Schnittpunkte mit Geraden grafisch ermitteln.

A9. Stellen Sie die Ebene mit der Parameterdarstellung

$$E: P = \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix} + s \cdot \begin{pmatrix} -0,5 \\ 0,5 \\ 2,5 \end{pmatrix} + t \cdot \begin{pmatrix} 2 \\ -1,5 \\ 0,5 \end{pmatrix} \quad (s, t \in \mathbb{R})$$

grafisch dar (nutzen Sie dafür eine transparente Textur). Stellen Sie außerdem den Aufpunkt und die beiden Richtungsvektoren der Ebene dar.

Fügen Sie der Grafik die Gerade hinzu, welche durch die Punkte  $A(-3; -1; 3)$  und  $B(-1; 1; -1)$  verläuft. Schätzen Sie so gut wie möglich die Koordinaten des Schnittpunktes der Geraden und der Ebene.

A10. Gegeben sind die drei Punkte  $P(-3; 3; 2)$ ,  $Q(3; -1; 3)$  und  $R(1; 3; -5)$ . Geben Sie eine Parametergleichung der Ebene durch  $P$ ,  $Q$  und  $R$  an.

Stellen Sie die drei Punkte, die ermittelten Richtungsvektoren und die Ebene grafisch dar. Ermitteln Sie aus der grafischen Darstellung so genau wie möglich die Koordinaten der Schnittpunkte der Ebene mit den Koordinatenachsen.

A11. Gegeben sind die Ebene  $E: P = \begin{pmatrix} 0 \\ 2 \\ -2 \end{pmatrix} + s \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + t \cdot \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}$  ( $s, t \in \mathbb{R}$ ) sowie die

Punkte  $A(0,5; -3; 1)$  und  $B(-1; -1; -7)$ . Überprüfen Sie, ob  $A$  und  $B$  auf der Ebene  $E$  liegen.

Grafiken zu den Aufgaben A10 und A11 sind auf S. 244 abgebildet.

<sup>50</sup>Mithilfe von `Parallelogram3d` erzeugte Objekte können (teil-)transparent dargestellt werden, was für `Surface`- und `Implicit3d`-Objekte in den bis 2007 aktuellen MuPAD-Versionen 3.1 und 4 nicht möglich ist. Es ist häufig sinnvoll, Ebenen teilweise transparent darzustellen, damit die von ihnen verdeckten Objekte erkennbar bleiben. In POV-Ray lassen sich im einfachsten Falle vordefinierte Texturen wie `rot_transp` einsetzen, siehe S. 335f. Bei Verwendung von MuPAD sind Anweisungen der Art `Color = RGB::Red.[0.5]` in Objektbeschreibungen einzufügen, womit eine Transparenz von 50% erreicht wird. Allerdings gelingt die transparente Darstellung einiger Objekttypen in MuPAD nicht.

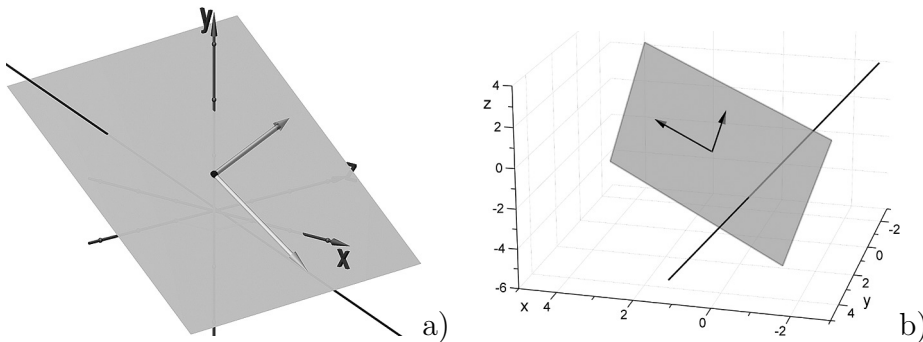


Abbildung 4.20:  
Lösungsgrafiken  
zu Aufgabe A9:  
a) in POV-Ray,  
b) in MuPAD

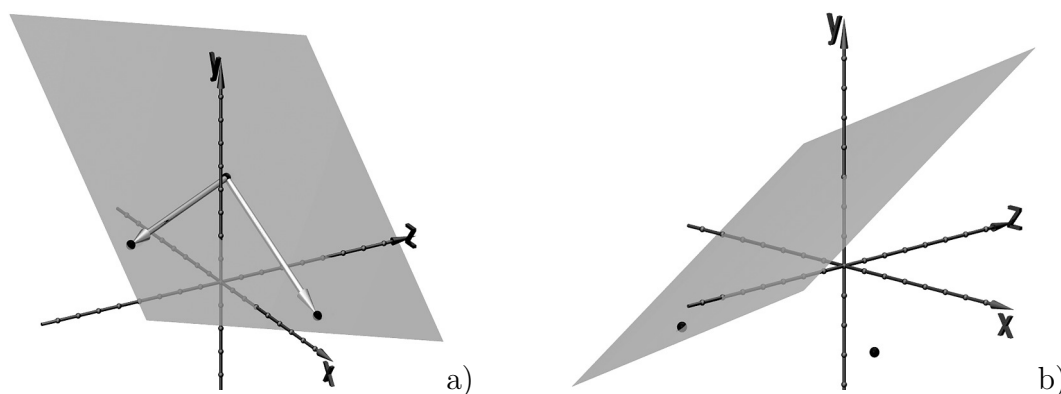


Abbildung 4.21: Grafiken zu den Aufgaben A10 (a) und A11 (b)

Um Schüler zu Schnittpunktberechnungen zwischen Ebenen und Geraden hinzuführen, ist es sinnvoll, in Anknüpfung an die Ermittlung von Spurpunkten (siehe Aufgabe A8 auf S. 241) zunächst sehr einfache Spezialfälle zu betrachten, die von den Schülern weitgehend selbstständig bearbeitet werden können.

A12. Gegeben ist die dreiseitige Pyramide  $ABCS$  mit  $A(2; 0; 2)$ ,  $B(10; 0; 6)$ ,  $C(2; 0; 12)$  und  $S(6; 8; 6)$ .

Bestimmen Sie alle Schnittpunkte der Kanten dieser Pyramide mit der Ebene, für deren sämtliche Punkte  $y = 2$  gilt. Stellen Sie die Ebene, die Kanten der Pyramide sowie die Schnittpunkte grafisch dar.

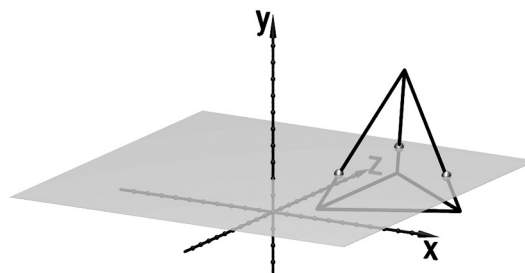


Abbildung 4.22: Grafik zu Aufgabe A12

Wie bereits erwähnt wurde, sollen die hier angegebenen Aufgaben nur exemplarisch die Nutzung einer Software wie MuPAD oder POV-Ray für die ergänzende Visualisierung von Standardaufgaben der (affinen) vektoriellen analytischen Geometrie des Raumes verdeutlichen.<sup>51</sup> Visuelle Schnittbetrachtungen zwischen Ebenen und die Darstellung von Ebenen, die durch implizite Gleichungen gegeben sind, wurden zunächst ausgeklammert, da diese innerhalb des nächsten Abschnitts – im Zusammenhang mit der Veranschaulichung linearer Gleichungssysteme – auftreten. Veranschaulichungen zum Skalarprodukt sowie zu Normalenvektoren und -gleichungen von Ebenen sind Gegenstände von Abschnitt 4.6.

<sup>51</sup>Visualisierungen in der beschriebenen Art können auch die Bearbeitung von Schulbuchaufgaben ergänzen. In Unterrichtsversuchen wurde die Erfahrung gemacht, dass viele Schüler freiwillig Grafiken zur Kontrolle ihrer (rechnerisch zu bearbeitenden) Hausaufgaben erstellten, siehe die Abschnitte 5.1 und 5.2.

#### 4.2.4 Schnittgeraden von Ebenen, Veranschaulichung linearer Gleichungssysteme und des Gauß-Algorithmus

Lineare Gleichungssysteme (LGS) sind bei der Bestimmung von Schnittmengen zwischen Geraden, Ebenen bzw. untereinander zwangsläufig zu lösen; ihre Behandlung ist daher in den Rahmenplänen Bestandteil des Stoffgebietes Analytische Geometrie oder eines speziellen vorgelagerten Stoffgebietes (vgl. 1.3). Um LGS und ihre Lösungsmengen sinnvoll geometrisch zu interpretieren, ist die Punktmengenauffassung von Geraden und Ebenen relevant.<sup>52</sup> Zunächst muss für die Schüler verständlich werden, dass alle Punkte, deren Koordinatentripel eine Gleichung der Form

$$Ax + By + Cz + D = 0$$

erfüllen, einer Ebene angehören. Dazu bietet es sich an, dass sie einige Lösungstripel entsprechender Gleichungen (z. B. durch Festlegen zweier der Variablen und jeweiliges Berechnen des Wertes der dritten Variablen) bestimmen und die zugehörigen Punkte grafisch darstellen (siehe Abb. 4.23) sowie für vorgegebene Lösungstripel grafisch und rechnerisch überprüfen, ob sie einer Ebene angehören.

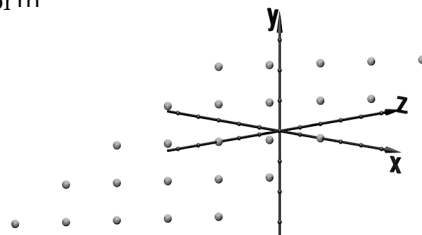


Abbildung 4.23: Punkte der Ebene mit der impliziten Gleichung  $1,5x + 2,5y - z - 3 = 0$

Ausgehend von diesen Überlegungen lässt sich herausarbeiten, dass Lösungsmengen von LGS, die aus zwei Gleichungen mit 3 Variablen bestehen, i. Allg. Geraden im Raum beschreiben. In diesem Zusammenhang ist es sinnvoll, Aufgaben zu stellen, bei denen die Schüler erkennen, in welchen Fällen die durch zwei Gleichungen gegebenen Ebenen parallel oder identisch sind.<sup>53</sup> Dazu können sie grafische Darstellungen anfertigen.<sup>54</sup>

A13. Ermitteln Sie grafisch, welche der durch folgende Gleichungen beschriebenen Ebenen parallel oder identisch sind. Stellen Sie dazu die Ebenen in verschiedenen Farben dar und notieren Sie die verwendeten Farben.<sup>55</sup>

$$\begin{array}{ll} E_1 : & 2x - 3y + 5z + 3 = 0 \\ E_2 : & -x + 1.5y - 2.5z + 5 = 0 \\ E_3 : & -2x - 2.5y + 3z + 6 = 0 \end{array} \quad \begin{array}{ll} E_4 : & -4x + 6y - 10z + 20 = 0 \\ E_5 : & 6x + 7.5y - 9z + 30 = 0 \end{array}$$

<sup>52</sup>Parameterdarstellungen von Geraden bzw. Kurven sind (wie bereits ausgeführt wurde, vgl. 3.3.2 und 4.2.2) m. E. derjenige Gegenstand innerhalb des Stoffgebietes Analytische Geometrie, anhand dessen sich der Punktmengengedanke am anschaulichsten herausarbeiten lässt. Eine Erweiterung der dabei angestellten Überlegungen auf Parameterdarstellungen von Ebenen wurde auf S. 242 angedeutet. Daran kann bei den hier geschilderten Betrachtungen zu Ebenen, die durch Gleichungen beschrieben sind, angeknüpft werden. Gleichzeitig ist der Bezug zu linearen Funktionen und einfachen Geradengleichungen, die in der SI behandelt wurden, herzustellen.

<sup>53</sup>Es wird davon ausgegangen, dass die Schüler Normalenvektoren von Ebenen noch nicht kennen.

<sup>54</sup>Zur Darstellung durch Koordinatengleichungen gegebener Ebenen können in POV-Ray bzw. MuPAD die auf S. 242 beschriebenen Anweisungen genutzt werden.

<sup>55</sup>Eine MuPAD-Grafik, die Schüler bei der Bearbeitung von Aufgabe A13 anfertigen können, zeigt Abb. 4.24 auf S. 246. Um bei Grafiken mit mehr als drei Ebenen einen Überblick zu gewinnen, muss zunächst ein geeigneter Blickwinkel gefunden werden.

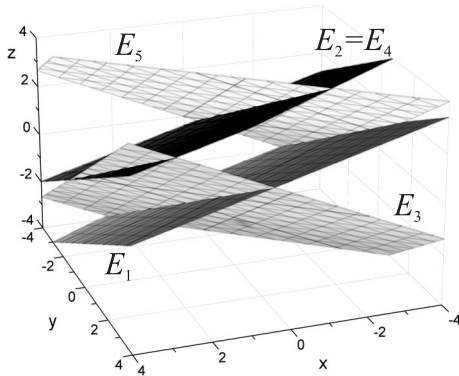


Abbildung 4.24:  
MuPAD-Grafik  
zu Aufgabe A13

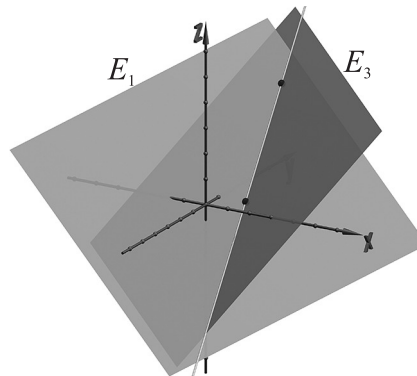


Abbildung 4.25:  
POV-Ray-Grafik  
zu Aufgabe A14

Nach einigen Aufgaben dieser Art werden viele Schüler Aussagen über die Koeffiziententripel bzw. -quadrupel paralleler und identischer Ebenen aufstellen können, auf deren Grundlage sich dann Lösungskriterien für aus zwei Gleichungen bestehende LGS erarbeiten lassen. Auch hierbei ist es sinnvoll, Bezüge zu entsprechenden Aussagen über LGS mit 2 Variablen, die in der SI behandelt wurden, herzustellen. Diese können genutzt werden, um Punkte von Schnittgeraden zweier Ebenen zu bestimmen, denn wird eine Variable festgehalten, so ergibt sich aus zwei Ebenengleichungen ein LGS mit 2 Variablen, für dessen Lösung die in der SI behandelten Verfahren ausreichend sind.<sup>56</sup>

A14. In Aufgabe A13 haben Sie festgestellt, dass die beiden Ebenen  $E_1$  und  $E_3$  genau die Punkte einer Geraden gemeinsam haben.

Ermitteln Sie die Koordinaten zweier Punkte, die jeweils zu beiden Ebenen gehören. Legen Sie dazu jeweils eine Koordinate fest (z. B.  $x = 0$  für den ersten und  $x = 1$  für den zweiten Punkt) und setzen Sie den Wert in die Gleichungen der Ebenen ein. Für jeden Punkt erhalten Sie jeweils ein Gleichungssystem mit 2 Gleichungen und zwei verbleibenden Variablen. Lösen Sie die beiden Gleichungssysteme und ermitteln Sie dadurch die fehlenden Koordinaten der beiden Punkte. Stellen Sie eine Parameterdarstellung der Schnittgeraden der beiden Ebenen  $E_1$  und  $E_3$  auf. Stellen Sie die beiden Ebenen, die von Ihnen bestimmten Punkte und die Schnittgerade in einer Grafik dar – kontrollieren sie Ihre Ergebnisse.<sup>57</sup>

Auf der Grundlage der bisher skizzierten Überlegungen dürfte für die Mehrheit der Schüler plausibel geworden sein, dass drei Ebenen, von denen jeweils zwei nicht parallel oder identisch sind, genau einen Punkt gemeinsam haben und somit die meisten LGS mit drei Gleichungen und drei Variablen genau ein Lösungstripel besitzen. Anhand des herausgearbeiteten Zusammenhangs zwischen der Parallelität von Ebenen und den Koeffiziententupeln der zugehörigen Gleichungen können die Schüler zudem entscheiden, ob ein LGS eindeutig oder mehrdeutig lösbar bzw. unlösbar ist und die „geometrische Struktur“ der Lösungsmenge (Punkt, Gerade, Ebene oder leere Menge) angeben.

<sup>56</sup>Erfahrungsgemäß müssen diese Verfahren allerdings wiederholt werden, da sich viele Schüler der SII nicht hinreichend daran erinnern können.

<sup>57</sup> Eine Lösungsgrafik zu dieser Aufgabe ist in Abb. 4.25 dargestellt. Die zugehörigen POV-Ray- bzw. MuPAD-Dateien, aus denen auch die Ergebnisse der Berechnungen hervorgehen, befinden sich in dem Ordner 4-2 (siehe Anhang D, S. 365).

### Visualisierungen des Gauß-Algorithmus

Zur Lösung von LGS können CAS oder geeignete Taschenrechner verwendet werden, mit denen mittlerweile in vielen Bundesländern im Mathematikunterricht gearbeitet wird. Ob und, wenn ja, mit welchem Gewicht der GAUSS-Algorithmus zur Lösung von LGS unter diesen Bedingungen noch behandelt werden sollte, ist umstritten. In den meisten, auch neueren, Rahmenplänen ist seine Behandlung jedoch vorgesehen (vgl. 1.3). Wenn der GAUSS-Algorithmus Gegenstand des Unterrichts der analytischen Geometrie ist, so sollte m. E. nicht nur Wert auf sein rechnerisches Beherrschen gelegt, sondern auch das Ziel verfolgt werden, dass die Schüler ein anschaulich-geometrisches Verständnis der Umformungsschritte erlangen. Im Folgenden werden zwei geometrische Interpretationen für eindeutig lösbare LGS mit drei Gleichungen und drei Variablen beschrieben.

Die am nächsten liegende (und sicherlich wichtigste) geometrische Interpretation des GAUSS-Algorithmus bezieht sich auf die Lage der drei Ebenen, die durch die Gleichungen des LGS gegeben sind. Die Koordinaten des Schnittpunktes dieser drei Ebenen (falls ein solcher existiert) bilden die Lösung des LGS. Da die durch den GAUSS-Algorithmus vorgenommen Umformungen äquivalent sind, können sich die Koordinaten des Schnittpunktes dabei nicht ändern. Diese Umformungen verändern die drei Gleichungen schrittweise so, dass Ebenen entstehen, die zunächst zu Koordinatenachsen und schließlich sogar zu Koordinatenebenen parallel sind, so dass sich die Koordinaten des Schnittpunktes unmittelbar aus den Gleichungen der drei Ebenen ablesen lassen. Sind alle drei durch das LGS gegebenen Ebenen zu jeweils einer Koordinatenebene parallel, so befindet sich das LGS in Diagonalf orm. Anhand eines ausgewählten LGS sind die einzelnen Umformungsschritte in den Abbildungen 4.26 und 4.27 dargestellt.<sup>58</sup>

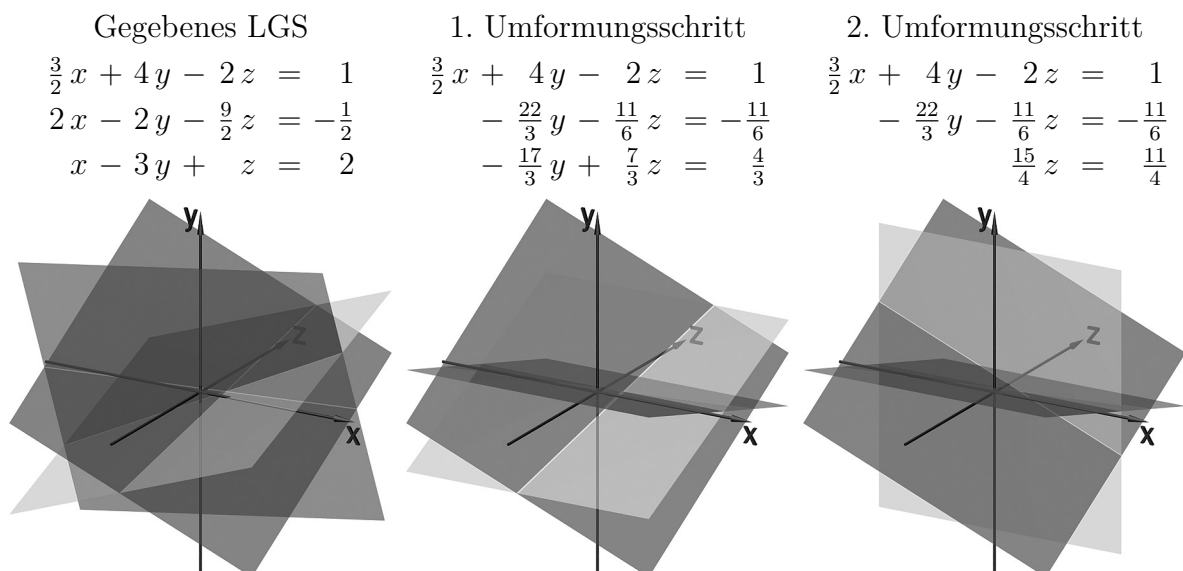


Abbildung 4.26: Visualisierung des GAUSS-Algorithmus durch Ebenen (Teil 1)

<sup>58</sup>Wesentlich besser erkennbar sind die Darstellungen in einer Präsentation, die sich in dem Ordner 4-2\Gauss-Algorithmus befindet und mit GaussAlg.exe oder gaussalg.htm gestartet wird. Sie enthält für jeden Schritt Videos, welche die Ebenen aus verschiedenen Blickrichtungen darstellen.

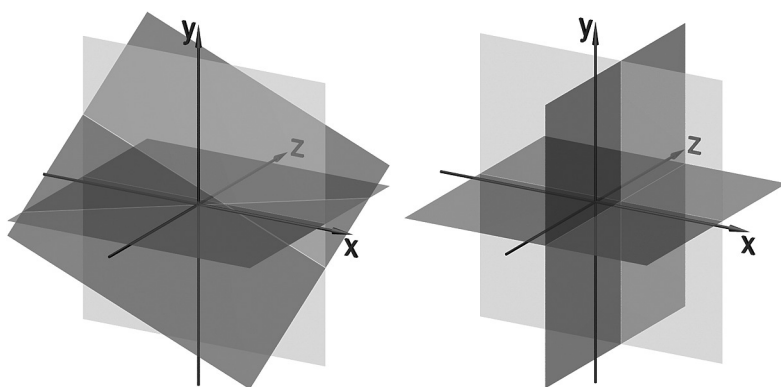


Abbildung 4.27: Visualisierung des GAUSS-Algorithmus durch Ebenen (Teil 2)

Eine weitere geometrische Interpretation des GAUSS-Algorithmus bezieht sich auf die Spaltenvektoren der Koeffizientenmatrix und den Vektor der absoluten Glieder.<sup>59</sup>

Die Frage nach den Lösungen eines LGS kann bei Betrachtung dieser Vektoren folgendermaßen gestellt werden:

*Gesucht sind die Koeffizienten, mit denen die Spaltenvektoren eines LGS zu multiplizieren sind, so dass der Vektor der absoluten Glieder eine Raumdiagonale des durch die derart multiplizierten Spaltenvektoren aufgespannten Parallelepipeds bildet.*

Das auf diese Weise gebildete Parallelepipet wird durch den GAUSS-Algorithmus schrittweise in einen Quader überführt, dessen Seiten auf den Koordinatenachsen liegen bzw. zu diesen parallel sind.

In Abb. 4.28 ist das auf S. 247 angegebene LGS durch seine Spaltenvektoren dargestellt. Entsprechende Darstellungen der daraus nach den Umformungsschritten des GAUSS-Algorithmus entstehenden LGS sowie zugehörige Videos enthält die bereits erwähnte Präsentation in dem Ordner 4-2\Gauss-Algorithmus.

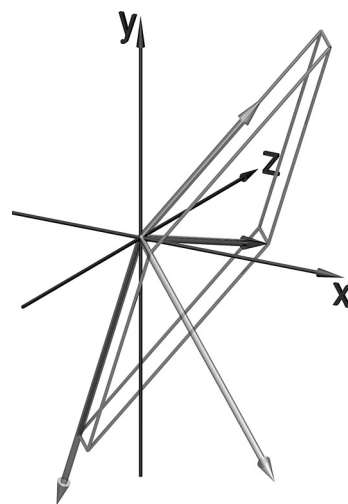


Abbildung 4.28: Spaltenvektoren eines LGS

Die Anfertigung komplexerer Visualisierungen ist zeitaufwändig, weshalb wahrscheinlich die meisten Schüler Darstellungen der Umformungsschritte des GAUSS-Algorithmus nicht selbst erstellen werden. Wenn sie jedoch vorher einige Ebenen und ihre Schnittgeraden dargestellt haben, so sind sie dazu prinzipiell in der Lage. Die hier beschriebenen Veranschaulichungen können sie sich zumindest ansehen sowie in vorbereiteten Dateien durch die Veränderung von Koeffizienten Visualisierungen für andere LGS erstellen.<sup>60</sup>

<sup>59</sup>Um diese Überlegungen anzustellen, müssen keine Matrizen behandelt werden; die Spaltenvektoren lassen sich direkt aus den LGS entnehmen, auch wenn diese wie auf S. 247 geschrieben sind.

<sup>60</sup>Neben den hier beschriebenen Visualisierungen stehen auch die POV-Ray-Dateien, mit denen sie erzeugt wurden, in den Materialien zu dieser Arbeit (siehe S. 365) sowie unter <http://www.afiller.de/3dcg> zur Verfügung. Die Berechnungen werden von POV-Ray selbst vorgenommen, so dass nur die Koeffizienten des gegebenen LGS verändert werden müssen, um die Umformungsschritte für andere LGS darzustellen. Evtl. muss dabei der Bildausschnitt (`intervall`) verändert werden. Bei sehr großen oder sehr kleinen Koeffizienten können allerdings Darstellungsprobleme auftreten.

## 4.3 Erstellung von Animationen durch die parameterabhängige Beschreibung von Kurven

### Einordnung und Ziele

Es wurde bereits mehrfach erwähnt, dass die Anfertigung kleiner Videos für viele Schüler ein hohes Motivierungspotenzial besitzt. Aus Abschnitt 2.6 geht hervor, dass Animationen durch die zeitabhängige Beschreibung von Positionen oder von Transformationen generiert werden können; es wird hier zunächst auf die erste dieser Herangehensweisen eingegangen.

Da die zeitabhängige Beschreibung von Positionen  $(x(t); y(t); z(t))$  mit der Parameterbeschreibung von (Bahn-)Kurven gleichzusetzen ist, kann die darauf basierende Erstellung von Animationen die Realisierung wichtiger *Unterrichtsziele* des Stoffgebietes Analytische Geometrie befördern. Dazu gehört insbesondere die Herausbildung einer dynamischen Sicht auf Parameterdarstellungen unter Betonung des funktionalen Zusammenhangs zwischen Parameterwerten (die als Zeitpunkte aufgefasst werden) und zugehörigen Punkten im Raum (siehe Abschnitt 3.3.2). Des Weiteren regt die Beschäftigung mit Bahnkurven zu Modellierungsüberlegungen an, bei denen Schüler zu gewünschten Bewegungsbahnen geeignete Parametergleichungen finden müssen. Derartige Überlegungen lassen sich mit „Experimenten“ in einer geeigneten Software kombinieren, indem die Schüler die durch bestimmte Parameterdarstellungen beschriebenen Kurven und die dadurch hervorgerufenen Animationen betrachten. Bezüge lassen sich zur Physik (speziell zur Kinematik) herstellen, indem Bewegungen mathematisch beschrieben und entsprechende Computersimulationen hergestellt werden.

Die Erstellung von Videos durch die zeitabhängige Beschreibung von Bewegungsbahnen kann an unterschiedlichen Stellen innerhalb eines Lehrgangs zur analytischen Geometrie thematisiert werden. So bietet sich die Behandlung dieses Themas unmittelbar im Anschluss an einen Einstieg in die räumliche Koordinatengeometrie nach Abschnitt 4.1 an, da vektorielle Vorgehensweisen nicht erforderlich sind und viele interessante Kurven durch parameterabhängige Koordinatenbeschreibungen dargestellt werden können. Soll die Vektorgeometrie einen hohen Stellenwert einnehmen, so bieten Parameterdarstellungen von Geraden einen geeigneten Anknüpfungspunkt (siehe 3.3.2 und 4.2.2).

### 4.3.1 Koordinatenbezogene Parameterdarstellungen – Kreise und Variationen von Kreisen

Die wichtigste Grundlage für Parameterdarstellungen von Kreisen in der Ebene haben die Schüler bereits in der SI kennengelernt – gewöhnlich werden dort die Sinus- und die Kosinusfunktion am Einheitskreis eingeführt; mit den in Abb. 4.29 verwendeten Bezeichnungen durch

$$\sin \alpha = y_\alpha, \quad \cos \alpha = x_\alpha.$$

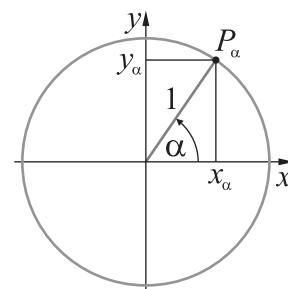


Abbildung 4.29: Trigonometrische Funktionen am Einheitskreis

Eine Verallgemeinerung auf Kreise in Mittelpunktslage mit beliebigem Radius  $r$  ist leicht möglich, woraus die Parameterdarstellung

$$x(\alpha) = r \cdot \cos \alpha, \quad y(\alpha) = r \cdot \sin \alpha; \quad \alpha \in [0; 2\pi)$$

eines Kreises der Ebene, dessen Mittelpunkt im Koordinatenursprung liegt, hergeleitet werden kann.<sup>61</sup> Parameterdarstellungen von Kreisen, die im Raum auf Koordinatenebenen oder dazu parallelen Ebenen liegen, ergeben sich daraus, indem eine der drei Koordinaten als Konstante dargestellt wird, z. B.  $z(\alpha) = c$ . Ausgehend von diesen Überlegungen können Schüler Animationen kreisförmiger Bewegungen generieren. Die dazu notwendigen Eingaben unter Verwendung von POV-Ray bzw. VPython wurden bereits in den Abschnitten 2.6.1 bzw. 2.6.5 beschrieben. Auch mithilfe der CAS Maple, Mathematica und MuPAD lassen sich Animationen erstellen. Eine Kugel mit dem Radius  $r_K$ , die sich auf einer Kreisbahn in einer zur  $x$ - $y$ -Ebene parallelen Ebene mit konstanter Winkelgeschwindigkeit bewegt, lässt sich in MuPAD z. B. durch

```
plot::Sphere(rk, [r*cos(t),r*sin(t),h], t = 0..2*PI, Frames=200)
```

beschreiben.<sup>62</sup>

Hinsichtlich der notwendigen mathematischen Überlegungen ist es unbedeutend, ob Schüler die Animation eines sich auf einer Kreisbahn bewegenden Objekts (z. B. einer Kugel) oder eine Kameraanimation erstellen, bei der sich die Sicht auf eine gesamte Szene verändert, siehe S. 153.<sup>63</sup> Erfahrungsgemäß sind Kameraanimationen für die Mehrzahl der Schüler interessanter. Um jedoch eine Vorstellung von zu erwartenden Animationsverläufen zu gewinnen, empfiehlt es sich, zunächst sichtbare Objekte zu animieren. Dies lässt sich leicht dadurch motivieren, dass auch für Schüler, die „Kamerafahrten“ erstellen möchten, damit der Ablauf von Animationen verständlicher wird und sie die Bahn sehen können, auf der sich die Kamera bewegen soll.

Die Darstellung von Bahnkurven ist in allen erwähnten Softwarepaketen möglich; in POV-Ray kann dies durch eine Vielzahl von Punkten bzw. kleinen Kugeln erfolgen, die durch eine `while`-Schleife generiert werden (siehe S. 237 und die Fußnote 156 auf S. 152). Ein Vorteil der Darstellung von Bewegungsbahnen durch diskrete Punkte besteht

<sup>61</sup>Durch die Addition von Mittelpunktskoordinaten lässt sich diese Parameterdarstellung leicht für beliebige Kreise in der Ebene verallgemeinern:  $x(\alpha) = r \cdot \cos \alpha + x_M$ ,  $y(\alpha) = r \cdot \sin \alpha + y_M$ . Überlegungen und Aufgaben zur Beschreibung von Kreisen, Spiralen und Lissajous-Figuren als Punktmengen in der Ebene beschrieb GIEDING in [72], Abschnitt 3.1.

<sup>62</sup>Beispieldateien mit in MuPAD erstellten Animationen (einschließlich einer Kameraanimation), aus denen auch die verwendeten Anweisungen hervorgehen, befinden sich in dem Ordner 4-3 (siehe Anhang D, S. 365f.). Nähere Informationen zu Animationen in MuPAD enthält [41], S. 245ff.

<sup>63</sup>Allerdings erfordern Objekt- und Kameraanimationen unterschiedliche geometrische Vorstellungen. Die Bewegung z. B. einer Kugel auf einer Kurve erschließt sich mit der Vorstellung von der Form der Kurve. Bewegt sich jedoch eine Kamera auf einer Kurve, so muss ein „Hineindenken“ in das bewegte Objekt und seine Sicht auf die Szene erfolgen – es ist in der Vorstellung also ein Wechsel des Bezugssystems nötig. Stellungswechsel zwischen der Sicht *auf* eine Kurve und der Sicht *von* der Kurve aus erfordern räumliches Denken im Sinne der von THURSTONE beschriebenen „Fähigkeit, mit räumlichen Vorstellungsinhalten gedanklich zu operieren“ (siehe S. 42).



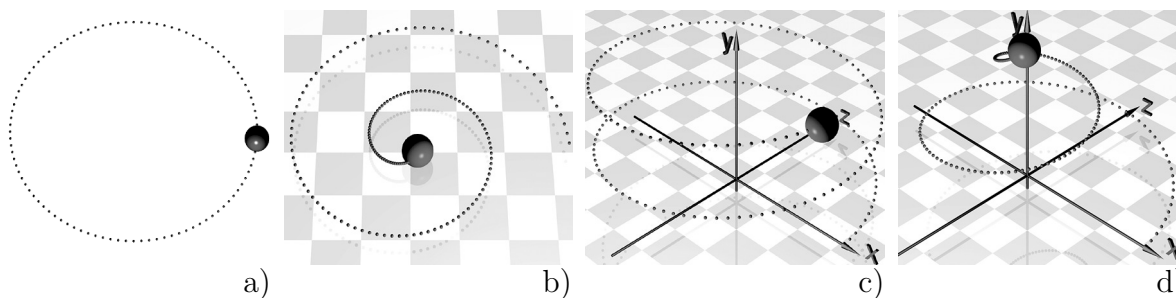


Abbildung 4.30: Bahnkurven: a) Kreis, b) Spirale, c) Schraubenlinie, d) konische Spirale

darin, dass ein visuelles Abschätzen der Bahngeschwindigkeit anhand der Abstände der Punkte möglich ist (siehe Abb. 4.30).

Nach der Behandlung der Parametergleichungen von Kreisen liegt es nahe, durch geeignete Veränderungen daran „verwandte“ Kurven parametrisch zu beschreiben. Dazu lassen sich folgende Fragen, die von den Schülern zu erwarten sind, aufgreifen:

1. Wie kann die Kamera um ein Objekt kreisen und sich diesem gleichzeitig nähern?
2. Wie lässt sich bei einer kreisförmigen Bewegung der Kamera gleichzeitig deren Höhe ändern, so dass Objekte aus unterschiedlichen Höhen betrachtet werden.<sup>64</sup>

Für die Realisierung der zuerst genannten Eigenschaft kann (zumindest mit Hilfen) von den Schülern herausgearbeitet werden, dass dazu die Konstante  $r$ , die für den Radius des zuvor betrachteten Kreises eingesetzt wurde, durch eine Funktion  $r(t)$  des sich zeitlich verändernden Parameters  $t$  ersetzt werden muss, z. B. durch  $(1 - t) \cdot r$ , falls sich der Abstand zum Mittelpunkt im Verlauf der Animation von  $r$  auf 0 verringern soll (für  $t \in [0; 1]$ ).<sup>65</sup> Durch diese Überlegung ergibt sich unmittelbar die Parameterdarstellung einer archimedischen Spirale (siehe S. 152).

Bei der Diskussion der Frage 2. können die Schüler erkennen, dass dazu die vorher konstant gehaltene dritte Koordinate durch eine Funktion  $z(t)$  bzw.  $y(t)$  des Parameters zu ersetzen ist. Wird dafür eine lineare Funktion gewählt (im einfachsten Falle  $y = t$  bzw.  $z = t$ , je nachdem, welche Koordinate bei der Beschreibung des Kreises konstant gehalten wurde), so entsteht aus der Kreisgleichung die Gleichung einer Schraubenlinie (S. 152). Durch die Kombination beider Überlegungen (parameterabhängige Beschreibungen des Radius und der „Höhe“ in der ursprünglichen Parameterdarstellung eines Kreises) ergibt sich bei Verwendung linearer Funktionen in  $t$  eine konische Spirale.<sup>66</sup>

Eine sinnvolle Veränderung gegenüber der Beschreibung des Kreises, von der ausgegangen wurde, betrifft die Terme, von denen der Sinus und der Kosinus gebildet werden. Bei

<sup>64</sup>Natürlich können beide Fragen auch so gestellt werden, dass sie den Verlauf von Kurven betreffen.

<sup>65</sup>Es ist im Zuge der hier beschriebenen Überlegungen sinnvoll, darauf einzugehen, dass die Verwendung normierter Parameterintervalle die Übersicht bei der zeitabhängigen Beschreibung mehrerer Eigenschaften erleichtert. Die Darstellung eines Kreises mittels  $x(\alpha) = r \cdot \cos \alpha$ ,  $y(\alpha) = r \cdot \sin \alpha$  mit  $\alpha \in [0; 2\pi)$  ist dazu durch  $x(t) = r \cdot \cos(2\pi t)$ ,  $y(t) = r \cdot \sin(2\pi t)$  mit  $t \in [0; 1]$  zu ersetzen.

<sup>66</sup>Siehe S. 152; konische Spiralen eignen sich recht gut als Bahnkurven für Kameraanimationen, wie das auf diese Weise entstandene Video `Kameraflug3.mpg` in dem Ordner 2-6 zeigt.

Spiralen und Schraubenlinien ist es oft erwünscht, mehr als eine Umdrehung zurückzulegen. Den Schülern könnte dazu die Frage gestellt werden, wie die Terme  $\cos(2\pi t)$  und  $\sin(2\pi t)$  verändert werden müssen, damit für  $t \in [0; 1)$  eine Spirale bzw. Schraubenlinie mit mehreren „Windungen“ entsteht, so dass sich z. B. die in Abb. 4.30 b), c) und d) dargestellten Kurven ergeben.

Weitere Variationen der bisher betrachteten Kurven ergeben sich aus der Verwendung nichtlinearer Funktionsterme in  $t$  für die Höhe bzw. den Radius. So kann die Aufgabe gestellt werden, die Parameterdarstellung der Schraubenlinie so zu verändern, dass sich deren Punkte zunächst sehr langsam und später schneller von denen des ursprünglich betrachteten Kreises entfernen.<sup>67</sup> Es bestehen vielfältige Möglichkeiten für funktionale Überlegungen, bei denen Schüler ausgehend von qualitativen Beschreibungen gewünschter Kurvenverläufe darüber nachdenken, durch welche Funktionsterme diese entstehen können, und ihre Überlegungen mithilfe der Software überprüfen. Durch Multiplikation der trigonometrischen Terme in den Parameterdarstellungen mit unterschiedlichen Faktoren (anstelle eines einheitlichen Radius) ist auch die Erzeugung von Ellipsenbahnen sowie Bahnen auf „elliptischen Spiralen“ und „elliptischen Schraubenlinien“ möglich.<sup>68</sup>

### 4.3.2 Vektorielle Parameterdarstellungen

Sollen im Unterricht vektorielle Beschreibungen und Vorgehensweisen im Vordergrund stehen, so bietet es sich an, in Parameterdarstellungen von Geraden den Parameter als Zeit zu interpretieren und auf dieser Grundlage auch einfache Animationen zu erstellen. Dabei kann unmittelbar an die in Abschnitt 4.2.2 beschriebenen Überlegungen zur Einführung von Parametergleichungen durch die Beschreibung und Darstellung von Punkten der entsprechenden Geraden angeknüpft werden. Unter Verwendung von Software wie POV-Ray oder eines der CAS Maple, Mathematica und MuPAD ist auf dieser Grundlage die Animation geradliniger Bewegungen sowie die Erstellung von Kameraanimationen möglich, bei denen sich die Kamera auf einer Geraden bewegt.<sup>69</sup> Derartige Animationen sind natürlich weniger interessant und attraktiv als Animationen unter Nutzung der im vorangegangenen Abschnitt beschriebenen Kurven, erfordern allerdings weniger Zeit und können die Behandlung der Parameterdarstellungen von Geraden ergänzen und dabei insbesondere funktionale Überlegungen anregen.

Bei Animationen erhalten Parameterdarstellungen einen Aspekt, der die geometrische Gestalt der durch sie beschriebenen Objekte nicht beeinflusst, nämlich die Geschwindigkeit von Bewegungen. So beschreiben z. B. die beiden Parameterdarstellungen

<sup>67</sup>Ebenso sind Variationen der archimedischen Spirale möglich, wobei jedoch nicht damit zu rechnen ist, dass Schüler durch eigene Experimente zur logarithmischen Spirale gelangen, sondern eher damit, dass sie quadratische, Potenz- und evtl. Wurzelfunktionen nutzen, um die zuvor verwendeten linearen Funktionen  $r(t)$  zu ersetzen.

<sup>68</sup>Der Ordner 4-3 enthält Beispiele für Bewegungen auf Ellipsenbahnen.

<sup>69</sup>Die technischen Vorgehensweisen hierbei unterscheiden sich nicht von denen, die im vorangegangenen Abschnitt sowie in 2.6.1 beschrieben wurden. Auch Beispiele für einfache lineare Animationen enthält der Ordner 4-3: **MuPAD-Animation-Geradenbahn**, **POVRay-Animation-Geradenbahn**.

$$\begin{aligned} (1) \quad & P(t) = P_0 + t \cdot \vec{a} \quad (t \in \mathbb{R}_+) \quad \text{und} \\ (2) \quad & P(t) = P_0 + t^2 \cdot \vec{a} \quad (t \in \mathbb{R}_+) \end{aligned}$$

dieselbe Halbgerade. Werden diese Parametergleichungen verwendet, um Animationen zu generieren, so ergibt (1) eine gleichförmige und (2) eine gleichmäßig beschleunigte Bewegung auf dieser Halbgeraden. In Abb. 4.31 ist dies durch die Abstände der Punkte erkennbar; zwischen zwei benachbarten Punkten verstreicht jeweils gleich viel Zeit.

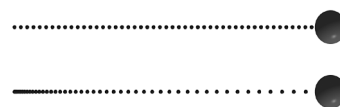


Abbildung 4.31: Gleichförmige (oben) und gleichmäßig beschleunigte Bewegung (unten)

Wie bereits dieses einfache Beispiel zeigt, lassen sich bei der Arbeit mit Animationen Verbindungen zum Physikunterricht herstellen, funktionale Aspekte durch die Betrachtung unterschiedlicher Funktionen  $f(t)$ , die den Zeitparameter ersetzen, vertiefen<sup>70</sup> sowie einfache Simulationen erstellen. In den meisten Bundesländern werden im Physikunterricht der SII Bewegungen vektoriell beschrieben, wobei der schräge Wurf Unterrichtsgegenstand ist. Dieser kann als eine aus einer gleichförmigen und einer gleichmäßig beschleunigten Bewegung zusammengesetzte Bewegung aufgefasst werden.

Als Summe einer in  $t$  linearen Komponente und des mit  $t^2$  multiplizierten Beschleunigungsvektors ergibt sich die Gleichung  $\vec{x} = \vec{x}_0 + \vec{v}_0 \cdot t + \frac{1}{2} \vec{g} \cdot t^2$  des schrägen Wurfs (welche die Wurfparabel beschreibt, siehe Abb. 4.32). Unter Verwendung einer geeigneten Software können Schüler den schrägen Wurf recht leicht simulieren (siehe z. B. den POV-Ray-Code auf S. 152f.) und auf analoge Weise Bahnen anderer zusammengesetzter Bewegungen untersuchen.

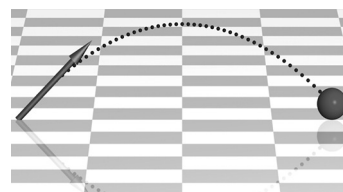


Abbildung 4.32: Schräger Wurf

Die Beispiele dieses Abschnitts zeigen, dass es mit recht elementaren mathematischen Mitteln, die an den Unterricht der SI anknüpfen, möglich ist, interessante Kurven zu modellieren und auf dieser Grundlage Animationen zu erstellen.<sup>71</sup> Dazu ist die Vektorrechnung nicht erforderlich, aber auch ausgehend von vektoriellen Geradengleichungen sind interessante Überlegungen zu Bewegungsbahnen möglich. Aus mathematikdidaktischer Sicht erscheint der beschriebene Gegenstandsbereich vor allem deshalb lohnenswert, weil die Beschäftigung damit anspruchsvolle Überlegungen zu funktionalen Zusammenhängen „anstößt“. Eine Möglichkeit der theoretischen Vertiefung unter Herstellung von Beziehungen zur Analysis wird im Folgenden skizziert.

<sup>70</sup>Auch für Kameraanimationen sind Beschleunigungs- und Abbremsvorgänge interessant, die Schüler mithilfe geeigneter Funktionen modellieren können. Dies beschränkt sich natürlich nicht auf geradlinige Bewegungen; auch bei Bewegungsbahnen auf beliebigen, durch Parametergleichungen beschriebenen, Kurven (wie bei den Beispielen auf S. 251) lassen sich mittels Substitution des Parameters  $t$  durch, auch stückweise definierbare, Funktionen  $f(t)$  Änderungen der Geschwindigkeit hervorrufen.

<sup>71</sup>Allerdings darf die dazu erforderliche Zeit nicht unterschätzt werden. Selbst für relativ elementar erscheinende funktionale Überlegungen, die z. B. von Kreisen zu Spiralen oder Schraubenlinien führen, brauchten auch Studierende in Seminaren recht lange – da sie diese Überlegungen als interessant empfanden und in Experimentierfreude verfielen, waren sie jedoch bereit, relativ viel Freizeit dafür aufzuwenden. Dies dürfte auch bei vielen Schülern der Fall sein, da die Erstellung interessanter Videos, wie mehrfach erwähnt wurde, überaus motivierend für Jugendliche ist.

### 4.3.3 Bestimmung von Bahngeschwindigkeiten auf gekrümmten Bewegungsbahnen als Möglichkeit der theoretischen Vertiefung in Leistungskursen

Wie bereits auf S. 252f. ausgeführt wurde, ist bei der Erzeugung von Animationen neben der Form von Bewegungskurven auch eine nichtgeometrische Eigenschaft von Bedeutung, nämlich der *Verlauf der Geschwindigkeit* der Bewegung. Unter der Geschwindigkeit wird die Ableitung der Ortsvektorfunktion (und somit der Parameterdarstellung) einer Kurve nach der Zeit verstanden:

$$\vec{v}(t) = \frac{d\vec{x}}{dt} = \begin{pmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \\ \frac{dz}{dt} \end{pmatrix}.$$

Für die Beschreibung von Animationen durch Parameterdarstellungen interessieren insbesondere die Beträge der Geschwindigkeiten zu verschiedenen Zeiten, also

$$v(t) = \frac{ds}{dt} = \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2 + \left(\frac{dz}{dt}\right)^2} = \sqrt{x'(t)^2 + y'(t)^2 + z'(t)^2}.$$

Diese „skalare Geschwindigkeit“ wird auch als Bahngeschwindigkeit bezeichnet, es handelt sich dabei um die Ableitung der zurückgelegten Bogenlänge nach der Zeit – da die Bogenlänge  $s$  zwischen zwei Punkten  $P(t_1)$  und  $P(t_2)$  einer Raumkurve durch

$$s = \int_{t_1}^{t_2} \left| \frac{d\vec{x}}{dt} \right| dt = \int_{t_1}^{t_2} \sqrt{x'(t)^2 + y'(t)^2 + z'(t)^2} dt$$

definiert ist, ergibt sich auch dadurch die o. a. Gleichung für  $v(t)$ .

Wie diese kurzen Betrachtungen zeigen, erfordern Berechnungen von Bahngeschwindigkeiten auf gekrümmten Bewegungskurven mathematische Überlegungen, die über die üblichen Unterrichtsinhalte des Stoffgebietes Analytische Geometrie erheblich hinausgehen, die Anwendung von Kenntnissen aus der Analysis auf parameterabhängige Koordinatenfunktionen erfordern und sich zu elementaren differenzialgeometrischen Betrachtungen (Bogenlänge von Kurven) fortsetzen lassen.<sup>72</sup> Visualisierungen der Bewegungskurven (insbesondere durch diskrete Punkte, die einen optischen Eindruck der Bahngeschwindigkeiten vermitteln, siehe Abb. 4.30, 4.31 und 4.32) ermöglichen es auch hierbei, anspruchsvolle mathematische Überlegungen durch Visualisierungen zu ergänzen und experimentelle Vorgehensweisen einzubeziehen. Allerdings dürften Aufgaben zur Bestimmung von Bahngeschwindigkeiten unter Verwendung der Ableitungen von Koordinatenfunktionen nach dem Parameter (der Zeit) sehr leistungsstarken Kursen oder einzelnen Schülern vorbehalten bleiben.

<sup>72</sup>Eine prinzipiell denkbare Anwendung der Betrachtungen zu Bahngeschwindigkeiten könnte darin bestehen, mithilfe einer Substitution des Parameters  $t$  durch eine Parameterfunktion  $f(t)$  eine konstante Bahngeschwindigkeit auf spiralförmigen Bewegungsbahnen zu erreichen. Die Bedingung  $v(f(t)) = c$  führt dann allerdings zu einer Differenzialgleichung, deren Lösung den Rahmen des Mathematikunterrichts – auch in Leistungskursen – überschreiten dürfte.

## 4.4 Geometrische Transformationen

### 4.4.1 Stellenwert und Arten der Beschreibung von Transformationen bei der Einbeziehung von Elementen der Computergrafik in das Stoffgebiet Analytische Geometrie

Affine Abbildungen, Koordinatentransformationen und Projektionen, ihre Darstellung durch Matrizen sowie darauf basierende Sichtbarkeitsbestimmungen gehören – wie aus Abschnitt 2.4 und dem Schema auf S. 84 hervorgeht – zu den zentralen mathematischen Grundlagen der Bilderzeugung in der 3D-Computergrafik.<sup>73</sup> Allerdings wurde bereits in Abschnitt 3.1 ausgeführt, dass Beschreibungen geometrischer Transformationen durch Matrizen für die praktische Arbeit mit Computergrafik-Software bzw. die Erstellung dreidimensionaler Darstellungen in CAS nicht unbedingt erforderlich sind und damit ein früher unterbreiteten Vorschlägen zugrunde liegender Aspekt ihrer Motivation weitgehend entfällt. Dies resultiert aus folgenden Eigenschaften moderner Software:

- Abbildungen dreidimensional beschriebener Objekte und Szenen in eine Projektionsebene nimmt die Software selbstständig vor. Der Benutzer kann die Projektion durch die Wahl von Kamera- und Blickpunktkoordinaten, des Öffnungswinkels sowie ggf. eines „up-Vektors“ beeinflussen, der bestimmt, welche Richtung des Raumes auf die vertikale Richtung des Bildes abgebildet wird. Zusätzlich besteht die Möglichkeit, zwischen einer Parallel- und einer Zentralprojektion zu wählen.<sup>74</sup>
- Die in der Computergrafik und -animation am häufigsten benötigten Abbildungen geometrischer Objekte<sup>75</sup> im Raum sind Verschiebungen, Drehungen und Skalierungen (Streckungen mit richtungsabhängigen Streckungsfaktoren), für die auch elementare Beschreibungen mittels `translate`, `rotate` und `scale` verfügbar sind.<sup>76</sup> Diese lassen sich erheblich leichter anwenden als Transformationsmatrizen. Abbildungen, deren Anwendung eine Beschreibung durch Matrizen tatsächlich erfordert (wie die in 2.4.1 beschriebenen Scherungen), kommen nur selten zum Einsatz.<sup>77</sup>

<sup>73</sup>Eine Ausnahme hinsichtlich der Projektionen bildet das Raytracing-Verfahren. Bei diesem Verfahren wird auf Lichtstrahlen, die von der Kamera ausgehen und durch festgelegte Punkte der Bildebene verlaufen, nach Schnittpunkten mit Objekten „gesucht“ (vgl. 2.5.5). Projektionen treten dabei nur indirekt auf – letztendlich erfolgt eine in entgegengesetzter Richtung vorgenommene Zentralprojektion; durch die Verfolgung paralleler Strahlen lässt sich auch eine Parallelprojektion erreichen.

<sup>74</sup>Die genannten Einstellungen können sowohl in POV-Ray als auch in MuPAD vorgenommen werden. Während jedoch in POV-Ray die Beschreibung einer Kamera für jede Szene notwendig ist, brauchen diese Angaben bei der Nutzung von MuPAD nicht gemacht zu werden, da CAS anhand der abgebildeten Objekte auch automatisch die notwendigen Einstellungen vornehmen.

<sup>75</sup>In mathematischem Sinne handelt es sich dabei um Abbildungen des gesamten Raumes auf sich. Da Verschiebungen, Drehungen und Streckungen allerdings i. Allg. nur auf einzelne Objekte angewendet werden, wird hier von „Abbildungen geometrischer Objekte“ gesprochen.

<sup>76</sup>In Abschnitt 2.4.1 wurden diese Befehle für POV-Ray beschrieben. In MuPAD stehen sie analog unter den Bezeichnungen `plot::Translate3d`, `plot::Rotate3d` und `plot::Scale3d` zur Verfügung.

<sup>77</sup>Die Anwendung von Transformationsmatrizen kann sinnvoll sein, um in ausgezeichnete Lage erstellte Objekte in vorgegebene Positionen zu überführen und auszurichten. In den Makros für die

Bei dem in der vorliegenden Arbeit verfolgten Ansatz, mathematische Grundlagen der 3D-Computergrafik in Kontexten zu thematisieren, in denen diese von den Schülern durch praktische Tätigkeiten nachvollzogen werden können und das Verständnis dieser Grundlagen die Möglichkeiten für die Erstellung von Grafiken und Animationen bzw. für deren „Perfektionierung“ erweitert, haben Beschreibungen von Abbildungen durch Matrizen daher nur eine untergeordnete Bedeutung. Insbesondere stellt die Aufnahme von Elementen der 3D-Computergrafik in den Unterricht m. E. *kein zwingendes Argument* für die Behandlung von Matrizen dar.

Aus den genannten Gründen wird auf Transformationsmatrizen im Folgenden nicht weiter eingegangen. Sollen im Unterricht geometrische Transformationen durch Matrizen beschrieben werden (wie es einige Rahmenpläne vorsehen, vgl. 1.3), so bestehen Möglichkeiten, dies mit der Erstellung computergrafischer Darstellungen zu verbinden, siehe [56], [108], [152], [153], [182], [266] und [280] sowie die diesbezüglichen Bemerkungen in Abschnitt 3.1.<sup>78</sup> Durchaus denkbar ist auch, dass einzelne Schüler, die für die theoretischen Grundlagen der Computergrafik Interesse aufbringen, sich mit der analytischen Beschreibung von Koordinatentransformationen und Projektionen auseinander setzen;<sup>79</sup> hierin besteht eine m. E. sinnvolle Möglichkeit innerer Differenzierung.

Die folgenden Ausführungen beziehen sich auf elementargeometrische Beschreibungen geometrischer Transformationen und sollen zeigen, dass auch auf dieser Grundlage interessante Überlegungen möglich sind, die von Schülern zudem weitgehend selbstständig geführt werden können.

#### 4.4.2 Parallel- und Zentralprojektionen

Wie bereits ausgeführt wurde, ist die analytische Beschreibung von Projektionen für die Erstellung von Computergrafiken nicht unabdingbar. Jedoch ist ein Verständnis für die Abbildungseigenschaften von Projektionen sinnvoll, wenn auf die Perspektive von Abbildungen Einfluss genommen werden soll. Ein solches Verständnis lässt sich über

---

Darstellung von Ebenen, die durch implizite Gleichungen bzw. Parameterdarstellungen gegeben sind, wurde diese Möglichkeit genutzt, um dünne Quader mit zu Koordinatenachsen parallelen Kanten mithilfe von Matrizen so zu drehen und zu verschieben, dass sie Ausschnitte der gegebenen Ebenen darstellen, vgl. Anhang A, S. 343ff. Die damit gelöste Aufgabe steht jedoch weniger bei der Generierung von Grafiken, sondern eher bei der Programmierung von Modulen wie der genannten Makros.

<sup>78</sup>LINDNER beschrieb in [163] u. a. ausführlich die Beschreibung geometrischer Transformationen (Spiegelungen, Skalierungen, Scherungen, Drehungen und deren Nacheinanderausführungen sowie Projektionen) durch Matrizen und die darauf basierende Erstellung von Grafiken unter Verwendung des CAS MuPAD. Die MuPAD-Befehle `plot::Translate3d`, `plot::Rotate3d` und `plot::Scale3d` sowie deren zweidimensionale Pendanten nutzte er dabei nicht. Um für die Darstellung dreidimensionaler Körper verschiedene Projektionsmatrizen anzuwenden, griff er auf die zweidimensionalen Grafikbefehle von MuPAD zurück. Die Veröffentlichung zeigt, dass matrizielle Beschreibungen von Abbildungen unter Verwendung eines CAS sehr gut behandelt und veranschaulicht werden können. Eine Motivation für deren Behandlung durch das Bedürfnis, Grafiken zu generieren und dabei Abbildungen zu verwenden, ist jedoch kaum möglich, da hierfür einfachere Vorgehensweisen zur Verfügung stehen.

<sup>79</sup>Eine in diese Richtung gehende Facharbeit eines Schülers mit Anmerkungen zu ihrer Entstehung enthält der Ordner `Schuelerarbeiten\Facharbeit-S-Haaf` (siehe Anhang D, S. 365).

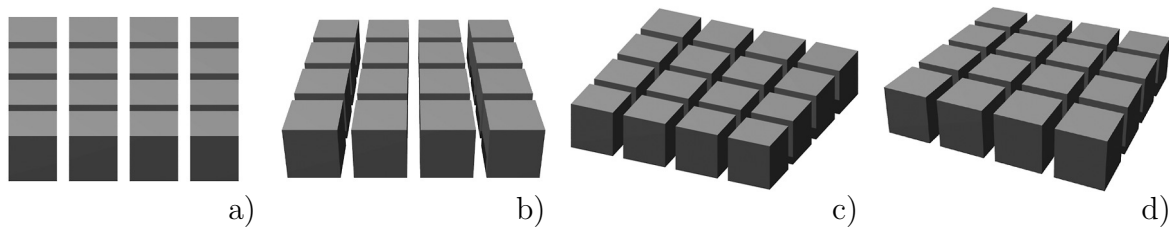


Abbildung 4.33: Szene mit 16 Würfeln, dargestellt aus zwei verschiedenen Kamerapositionen in Parallelprojektion (a, c) und Zentralprojektion (b, d)

elementargeometrische Betrachtungen erreichen, die teilweise Gegenstand des Unterrichts der SI sind.<sup>80</sup> In diesem Zusammenhang können die Einstellungen der Kamera thematisiert und deren Bezüge zu Projektionsebene und Zentrum sowie zu dem Teil des Raumes herausgearbeitet werden, der auf den in der berechneten Grafik sichtbaren (rechteckigen) Ausschnitt der Projektionsebene abgebildet wird (siehe 2.4.3). Es bietet sich dabei an, eine einfache Szene (die mehrere parallele Kanten und kongruente Objekte enthalten sollte) mit verschiedenen Kameraabständen und Öffnungswinkeln sowie vergleichsweise in Parallel- und Zentralprojektion darzustellen,<sup>81</sup> siehe z. B. Abb. 4.33. Anhand derartiger Abbildungen lassen sich Eigenschaften wie die Parallelentreue sowie die Form des abgebildeten Raumausschnittes bei den beiden Projektionsarten vergleichen. Diese Überlegungen können durch Versuche zur Sichtbarkeit innerhalb des Sichtquaders einer Parallelprojektion befindlicher, der Kamera naher sowie weit entfernter Objekte bei entsprechenden Zentralprojektionen ergänzt werden.

### 4.4.3 Translationen, Drehungen und Skalierungen

Die Anwendung von Translationen, Drehungen und Streckungen (teilweise mit unterschiedlichen Streckfaktoren entlang der verschiedenen Koordinatenachsen) ist insbesondere bei der Erstellung etwas umfangreicherer Szenen aus folgenden Gründen nötig bzw. zumindest im Sinne der Arbeitserleichterung sinnvoll:

- Durch Gleichungen bzw. Parameterdarstellungen gegebene Flächen können nur in spezieller Lage bezüglich des Koordinatensystems relativ einfach beschrieben werden. In POV-Ray lassen sich außerdem einige Objekte (wie Tori, Prismen und Rotationskörper) nur in ausgezeichneter Lage erzeugen, siehe 2.3.3 und 2.3.7. In diesen Fällen sind Verschiebungen und oft auch Drehungen erforderlich, um die gewünschte Anordnung innerhalb einer Szene zu erreichen.

<sup>80</sup>Interessante didaktische Vorschläge zu Projektionen und perspektivischen Darstellungen unter Nutzung eines DGS in der Klassenstufe 9 unterbreitete ELSCHENBROICH in [55].

<sup>81</sup>Das Raytracing-Verfahren wurde in Abschnitt 2.5.5 für den Fall der Zentralprojektion beschrieben, da dieser in der fotorealistischen Computergrafik fast immer auftritt. Allerdings lassen sich auch mit Raytracing-Programmen Parallelprojektionen erzeugen. Dazu werden statt Lichtstrahlen, die von der Kamera ausgehen, Strahlen betrachtet, die zu der Geraden durch die Kamera und den Mittelpunkt des Rechtecks, welches das zu berechnende Bild darstellt, parallel sind. In POV-Ray lassen sich Parallelprojektionen durch Einfügen von `orthographic` in die Definition der Kamera erreichen.

- Bereits in Abschnitt 4.1.2 wurde ausgeführt, dass bei der Modellierung zusammengesetzter Objekte die Ausrichtung entlang von Koordinatenachsen oder in Koordinatenebenen die Ermittlung geeigneter Koordinaten erleichtert. Um die zusammengesetzten Objekte dann innerhalb einer Szene zu positionieren und auszurichten, müssen Transformationen auf sie angewendet werden.
- Mitunter sollen mehrere Exemplare eines Objekts in einer Szene vorhanden sein und evtl. unterschiedliche Ausrichtungen und Größen haben. In diesem Falle ist es sinnvoll, ein Objekt nur einmalig zu erstellen und entsprechende Kopien (Instanzen) zu verschieben, zu drehen und ggf. zu skalieren.

Die Beschreibung von Transformationen durch Translationsvektoren, Drehwinkel und Streckfaktoren ist prinzipiell sehr einfach, es treten hierbei für Schüler der SII keine neuen mathematischen Inhalte auf – allerdings erfordert die Wahl der richtigen Größen (insbesondere der Drehwinkel) gründliche raumgeometrische Überlegungen, die das Raumvorstellungsvermögen fordern und anregen (soweit dies in der SII noch möglich ist, vgl. hierzu 1.4.4). Ein Beispiel der Nutzung von Transformationen für die mehrfache Verwendung von Objekten zeigt Abb. 4.34 a), die von einem Schüler einer 12. Klasse angefertigt wurde (siehe Abschnitt 5.2). Nachdem er einen Schneemann erstellt, ihn in eine Kugel eingeschlossen und ihm mittels `#declare schneekugel` einen Objektnamen gegeben hatte, fügte er seiner Szene mehrere Kopien hinzu:<sup>82</sup>

```
// Drei kleine Schneemänner auf dem Regal
object { schneekugel rotate <0,-40,0> scale <0.4,0.4,0.4> translate <-30.7,27.5,1>}
object { schneekugel rotate <0,-90,0> scale <0.4,0.4,0.4> translate <-31.1,27.5,22>}
object { schneekugel rotate <0,+40,0> scale <0.4,0.4,0.4> translate <-31,27.5,-26>}
```

Weitere Beispiele der Anwendung von Translationen und Drehungen für die Nutzung mehrerer Kopien komplexerer geometrischer Objekte zeigen die Abbildungen 4.34 b)

<sup>82</sup>Zu diesem anspruchsvollen Beispiel (siehe auch die farbige Abbildung B.35 auf S. 360 sowie die Datei `Schüler 7.2.pov` in dem Ordner `Schuelerarbeiten\Schneemaenner-IGH-Heidelberg`) ist anzumerken, dass der Lehrer des Kurses die Nutzung von Transformationen im Unterricht nicht angesprochen hatte und der betreffende Schüler selbst die Idee entwickelte, diese Möglichkeit zu nutzen.

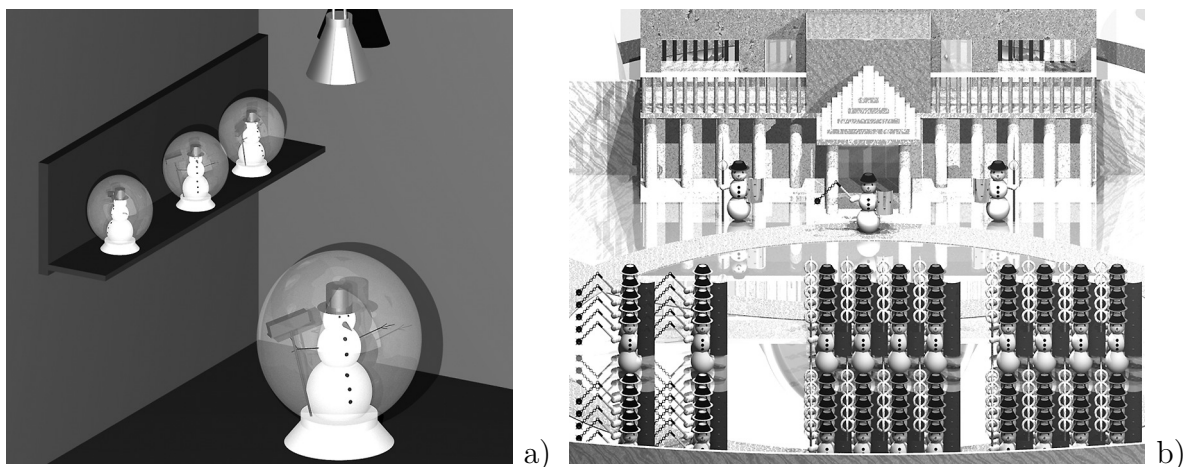


Abbildung 4.34: Vervielfachen von Objekten mithilfe von Transformationen



und 4.35, die von Studierenden in einem Seminar zur geometrischen Modellierung in der Computergrafik angefertigt wurden. Durch die Anwendung verschiedener Translationsvektoren auf wenige „Prototypen“ von Schneemännern generierte ein Student eine ganze „Schneemannarmee“, während eine Studentin ein Blatt als durch eine Splinekurve begrenztes Prisma (vgl. 2.3.7) konstruierte und durch geeignete Translationen und Drehungen mehrfach in ihrer Szene verwendete.<sup>83</sup>

Einer wichtiger Aspekt bei der Anwendung von Transformationen auf Objekte ist die *Nichtkommutativität* ihrer *Nacheinanderausführung*, die dazu zwingt, über die notwendige Reihenfolge von Verschiebungen, Drehungen und Skalierungen nachzudenken (siehe Abschnitt 2.4.1). Auch bei der Erzeugung von Animationen durch die zeitabhängige Beschreibung von Transformationen kommt diesem Aspekt eine wichtige Bedeutung zu, wie anhand zweier im folgenden Abschnitt beschriebener Aufgaben deutlich wird.

#### 4.4.4 Erzeugung von Animationen durch zeitabhängige geometrische Transformationen

Von einem Zeitparameter abhängige Beschreibungen von Translationen, Drehungen und ggf. auch Streckungen ermöglichen es, vielfältige Animationen zu erstellen und sind deshalb für viele Schüler interessant.<sup>84</sup> Sie müssen dabei vor allem Überlegungen zur geeigneten Synchronisation mehrerer Transformationen anstellen und die richtige Reihenfolge bei der Nacheinanderausführung mehrerer Abbildungen wählen. Hierzu kann Schülern z. B. die folgende Aufgabe gestellt werden:

- Bilden Sie die Drehung der Erde um sich selbst, des Mondes um die Erde und des Mondes um sich selbst in einer Animation nach. Berücksichtigen Sie dabei die Größenverhältnisse von Erde und Mond sowie die Umlaufzeiten. Den Abstand

<sup>83</sup>Die von den Studierenden erstellten zugehörigen POV-Ray-Dateien befinden sich in dem Ordner *Studierenden-Arbeiten* (siehe S. 365).

<sup>84</sup>In einem von F. RIEPER durchgeführten Unterrichtsprojekt, in dem die Schüler sich selbst für Schwerpunkte entscheiden konnten, generierten viele Schüler Animationen durch die zeitabhängige Beschreibung von Verschiebungen und Drehungen, siehe Abschnitt 5.3 und die Beispiele in dem Ordner *Schuelerarbeiten\Projekt-FPG-Cottbus-ma13-F-Rieper*.

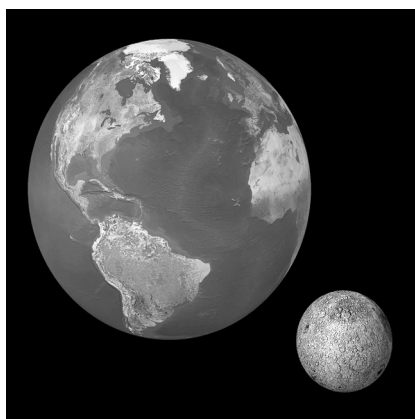


Abbildung 4.35: Gedrehte und verschobene Objekte (Blätter)

Abbildung 4.36: Bewegungen von Mond und Erde (Momentaufnahme)

des Mondes zur Erde können Sie allerdings nicht maßstabsgetreu nachbilden, da hierbei Erde und Mond viel zu klein dargestellt würden.

Bei der Lösung dieser Aufgabe ist hinsichtlich der Bewegung des Mondes darauf zu achten, dass (bei im Koordinatenursprung befindlichem Mondmittelpunkt) zunächst eine Drehung (Drehung des Mondes um seine eigene Achse), danach eine Translation und schließlich erneut eine Drehung (Mondbewegung um die Erde) ausgeführt werden müssen. Es ist jedoch i. Allg. nicht notwendig, dies mit Schülern ausführlich im Voraus zu diskutieren, da diese bei der Arbeit selbst erkennen, wenn sich aus einer falschen Reihenfolge der Nacheinanderausführung ein unerwünschtes Resultat ergibt und sich dann, wenn sie selbst keine geeigneten Lösungen finden, an den Lehrer wenden.<sup>85</sup>

Die folgende Aufgabe ist vor allem aus dem Grunde interessant, dass die Schüler bei ihrer Lösung genaue Überlegungen zur Synchronisation einer Translation und einer Drehung anzustellen haben.

- Lassen Sie eine (nicht einfarbige) Kugel oder einen Zylinder über eine Grundfläche rollen. Achten Sie darauf, dass die Kugel bzw. der Zylinder dabei nicht „rutscht“, sondern exakt rollt.



Zur Lösung dieser Aufgabe müssen die Schüler den Umfang eines Großkreises der Kugel berechnen und die Translationsgeschwindigkeit so wählen, dass die Kugel während einer Drehung um  $360^\circ$  genau um diesen Umfang verschoben wird (siehe 2.6.2, S. 154).<sup>86</sup>

Sehr viel Zeit ist für geometrische Transformationen im Unterricht nicht erforderlich, wenn diese lediglich in der hier beschriebenen elementaren Weise thematisiert werden. Es reichen kurze Hinweise auf diese Thematik, die auch innerhalb anderer Inhaltsbereiche erfolgen können. Aufgrund bereits kurz erwähnter Unterrichtserfahrungen ist m. E. damit zu rechnen, dass viele Schüler bei der Arbeit an eigenen Projekten Transformationen einsetzen und dabei über die Festlegung sowie die Animation geeigneter Drehwinkel und Translationsvektoren nachdenken.<sup>87</sup>

<sup>85</sup>Die Lösung der Aufgabe wurde bereits in Abschnitt 2.6.2 beschrieben. Ein entsprechendes Video **Erde+Mond-Rotation.mpg** und die zugehörige POV-Ray-Datei enthält der Ordner 2-6; Abb. 4.36 zeigt eine „Momentaufnahme“ aus dem Video.

<sup>86</sup>Ein der Aufgabe entsprechendes Video **Rollende-Kugel.mpg** und die zu seiner Herstellung angefertigten POV-Ray-Dateien enthält der Ordner 2-6.

<sup>87</sup>Wie bereits erwähnt wurde, entschieden sich viele Schüler in einem Unterrichtsprojekt für diesen Schwerpunkt und erreichten dabei interessante Ergebnisse. Eine Schülerin fertigte eine Animation mit einer Vielzahl von zeitabhängigen Transformationen an, die sich in dem Ordner **Schuelerarbeiten\Projekt-FPG-Cottbus-ma13-F-Rieper\Juliane-Mittelalter** befindet. Ähnliche Erfahrungen wurden bei der Arbeit mit Studierenden gesammelt, von denen viele die Erstellung einer Animation als Schwerpunkt eines Seminarprojekts wählten. Beispiele befinden sich in dem Ordner **Studierenden-Arbeiten**; hervorgehoben sei von diesen Arbeiten u. a. die Animation in dem Unterordner **Studierenden-Arbeiten\Lipp+Hainbuch\_MenschAerger**.

## 4.5 Visuell gestützte Untersuchung von Kegelschnitten, Flächen 2. Ordnung und weiteren Flächen

### 4.5.1 Räumlich-visuelle Einführung der Kegelschnitte

Zur Behandlung der Kegelschnitte wurden in der Literatur vielfältige Vorschläge unterbreitet und Konzepte entwickelt (vgl. 1.4.5), wobei insbesondere [235] einen umfassenden Überblick über die Thematik gibt. Im Rahmen der vorliegenden Arbeit soll lediglich auf Bezüge zwischen der Einbeziehung von Elementen der Computergrafik in den Unterricht und der Behandlung von Kegelschnitten eingegangen und ein dabei möglicher Zugang skizziert werden, der sich in Anknüpfung an die in Abschnitt 4.1 geschilderten Überlegungen zur Herangehensweise an das Stoffgebiet Analytische Geometrie anbietet.

Die folgenden Überlegungen gehen davon aus, dass der Einstieg in das Stoffgebiet koordinatengeometrisch und unter Verwendung einer Software wie POV-Ray oder eines CAS erfolgt ist. Wurden dabei die in Abschnitt 4.1 geschilderten Überlegungen im Unterricht thematisiert, so sind für eine Behandlung der Schnittfiguren eines Kreiskegels und einer Ebene bereits folgende Voraussetzungen gegeben:

- Den Schülern ist bekannt, dass die Wahl einer geeigneten Lage von Objekten bezüglich des Koordinatensystems Positionierungen vereinfachen kann.
- Sie haben Überlegungen zu Schnittfiguren von Kugeln und daraus zusammengesetzten Objekten mit Koordinatenebenen angestellt.
- Sie haben Gleichungen geometrischer Grundkörper (Kugeln, Kegel) hergeleitet.<sup>88</sup>
- Die Schüler können eine Software für die Darstellung räumlicher Objekte nutzen.

Auf der Grundlage dieser Voraussetzungen ist ein unmittelbarer Einstieg in die Thematik der Kegelschnitte möglich,<sup>89</sup> der z. B. mittels der folgenden Aufgabe erfolgen kann:

- *Erzeugen Sie einen Doppelkegel und eine Ebene, die diesen schneidet. Betrachten Sie die entstehenden Schnittfiguren zwischen dem Doppelkegel und der Ebene und verändern Sie die Position der Ebene durch Verschieben und Drehen. Versuchen Sie, so viele unterschiedliche Schnittfiguren wie möglich zu finden. Untersuchen Sie vor allem verschiedene Winkel zwischen der Kegelachse und der Ebene und vergleichen Sie diese Winkel jeweils mit dem Öffnungswinkel des Kegels.*<sup>90</sup>

---

<sup>88</sup>Zur Herleitung der Gleichung eines Kegels in spezieller Lage siehe S. 223; sollte die Herleitung in dem dort beschriebenen Kontext noch nicht erfolgt sein, so kann sie auch bei der Behandlung der Kegelschnitte durchgeführt werden.

<sup>89</sup>Für die Ermittlung von Gleichungen für Kegelschnitte auf der Grundlage des hier beschriebenen Ansatzes ist es außerdem notwendig, dass die Schüler Ebenen durch Gleichungen beschreiben können.

<sup>90</sup>Sollten die Schüler in der verwendeten Software noch keine Ebenen dargestellt haben, so ist an dieser Stelle darauf einzugehen. Besteht Zeitknappheit oder haben die Schüler zuvor nur wenig mit der Software gearbeitet, so können Vorlagen zur Verfügung gestellt werden, in denen sie Ebenen durch Eingabe weniger Werte (Winkel der Ebene zu einer Koordinatenachse, Schnittpunkt mit dieser Achse) erstellen und verändern. Entsprechende Vorlagen für POV-Ray enthält der Ordner 4-5. Sinnvoller ist es jedoch m. E., wenn Schüler selbst Szenen anfertigen.

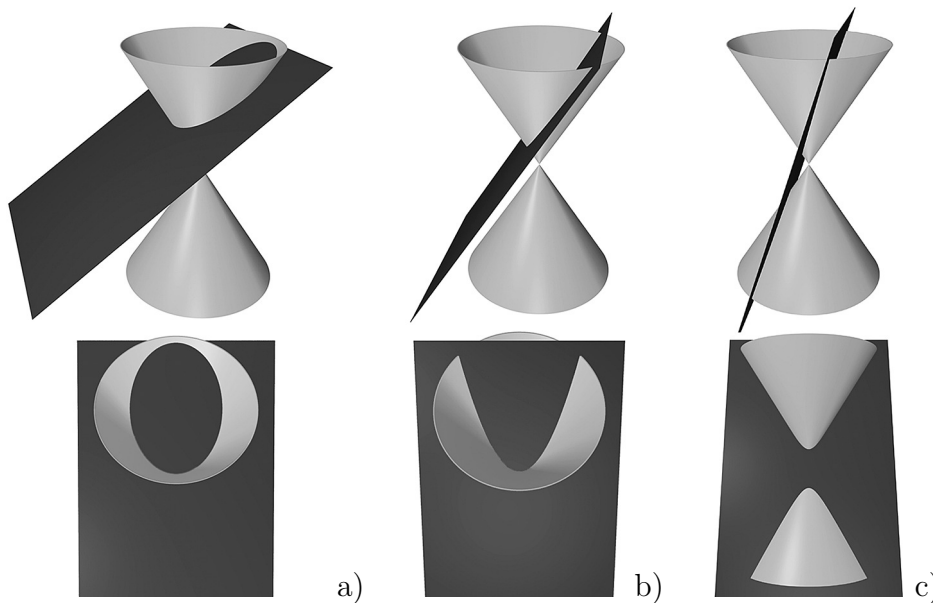


Abbildung 4.37:  
Schnitte einer  
Ebene und eines  
Kreiskegels aus  
jeweils zwei Ka-  
merapositionen

- a) Ellipse  
b) Parabel  
c) Hyperbel

Es ist für die Lösung dieser Aufgabe nicht notwendig, dass die Schüler Winkel zwischen Geraden und Ebenen berechnen können, da sie Ebenen zunächst in einer Koordinatenebene generieren und dann um eine Koordinatenachse drehen können. Falls die Kegelachse auf einer Koordinatenachse angeordnet wird, so ergeben sich die Winkel zwischen der Kegelachse und der Ebene aus dem Drehwinkel und können mit dem halben Öffnungswinkel des Kegels verglichen werden.

Einige Grafiken, die bei der Lösung der Aufgabe entstehen können, zeigt Abb. 4.37.<sup>91</sup> Bei der Bearbeitung der Aufgabe ist es für die Schüler leicht möglich, zu erkennen, dass beim Schnitt eines Kegels<sup>92</sup> mit einer Ebene geschlossene und offene Kurven (die teilweise aus zwei getrennten „Ästen“ bestehen) sowie für den Fall, dass die Ebene durch die Kegelspitze verläuft, Geraden, Paare sich schneidender Geraden bzw. ein einzelner Punkt entstehen können. Zwei Aspekte werden jedoch oft nicht ohne weitere Überlegungen wahrgenommen:

- Dass es sich bei den entstehenden geschlossenen Kurven tatsächlich um Ellipsen, d. h. um Kurven mit zwei Symmetrieachsen handelt, erkennen Schüler oft nicht und nehmen stattdessen an, dass diese Schnittkurven „eiförmig“ sind, also nur eine Symmetrieachse besitzen. Um zu verdeutlichen, dass tatsächlich Ellipsen entstehen, sollten zusätzlich zu der verwendeten Software herkömmliche räumliche Modelle herangezogen werden.<sup>93</sup>

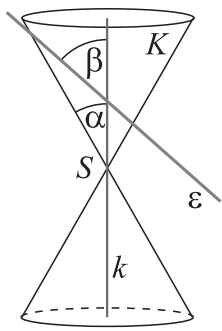
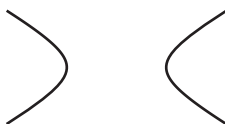

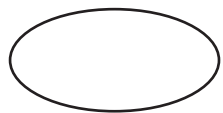
<sup>91</sup>Auf S. 361 sind außerdem Abbildungen zu Kegelschnitten enthalten, die zwei Schüler eines Grundkurses ma-13 erstellten. Weitere von diesen Schülern angefertigte Abbildungen und ein einfaches Video eines Kegelschnittes enthält der Ordner **Schuelerarbeiten\Projekt-FPG-Cottbus-ma13-F-Rieper\Tobias-Steffen-Kegelschnitte** (siehe Anhang D, S. 365). In den Ordnern 4-5 und 2-6 befinden sich außerdem zwei weitere Videos, welche die Übergänge der verschiedenen Schnittkurven zeigen.

<sup>92</sup>Die Bezeichnung „Kegel“ wird im Folgenden für einen in beide Richtungen der Kegelachse unendlich ausgedehnten geraden Kreiskegel verwendet.

<sup>93</sup>Auch Experimente mit dem Lichtkegel einer Taschenlampe sind hierbei sinnvoll. Ergänzend kön-

- Häufig wird übersehen, dass bei der Veränderung des Neigungswinkels der Schnittebene beim Übergang zwischen Ellipsen und Hyperbeln noch eine dritte Art von Kurven (Parabeln) erzeugt wird. Dies liegt vor allem daran, dass aufgrund der endlichen Ausdehnung grafischer Darstellungen oft nur einzelne Hyperbeläste als Schnittkurven sichtbar sind, die sich in ihrer Form wenig sichtbar von Parabeln unterscheiden. Es ist also notwendig, Schüler zum Nachdenken darüber anzuregen, dass – falls der halbe Öffnungswinkel des Kegels gleich dem Winkel zwischen der Ebene und der Kegelachse ist – eine besondere Schnittkurve entsteht, die sich von den in den anderen Fällen entstehenden Kurven unterscheidet.

Nachdem die Schüler „experimentell“ die möglichen Schnittfiguren zwischen einem Kegel und einer Ebene gefunden haben, sollten diese systematisch in Abhängigkeit von dem Winkel zwischen der Ebene und der Kegelachse zusammengefasst und ihre Formen verglichen werden, wobei eine Übersicht folgender Art entstehen kann:

	$\beta < \alpha$	$\beta = \alpha$	$\beta > \alpha$	
$S \in \varepsilon$	2 sich schneidende Geraden	Gerade	Punkt	
$S \notin \varepsilon$	Hyperbel 	Parabel 	Ellipse (Kreis für $k \perp \varepsilon$ ) 	
$\varepsilon$ – Ebene; $S$ – Spitze des Kegels; $k$ – Achse des Kegels; $\alpha$ – halber Öffnungswinkel des Kegels; $\beta$ – Winkel zwischen der Achse des Kegels und der Ebene				

Nach einer derartigen Zusammenfassung bietet es sich an, Gleichungen für die nicht entarteten Schnittkurven herzuleiten, um diese weiter zu untersuchen. Dabei kommen verschiedene Vorgehensweisen infrage, die in der Literatur ausführlich beschrieben sind.<sup>94</sup> Vergleichsweise sehr einfach ist es, für Spezialfälle der gegenseitigen Lage von Kegel und Ebene Gleichungen spezieller Ellipsen, Hyperbeln und Parabeln herzuleiten. So lassen sich die Gleichungen der Schnittkurven des speziellen Kegels mit einem Öffnungswinkel von  $45^\circ$ , der  $z$ -Achse als Kegelachse und der Spitze im Koordinatenursprung

$$K: x^2 + y^2 - z^2 = 0$$

mit den Ebenen, die durch die Gleichungen

nen die Schnittfiguren in der verwendeten Software aus verschiedenen Richtungen betrachtet werden. Außerdem empfiehlt sich die vergleichende Betrachtung von Zylinderschnitten, deren elliptische Form unmittelbar auf der Hand liegt, um festzustellen, dass dabei gleichartige Kurven entstehen.

<sup>94</sup>Eine der gebräuchlichsten Vorgehensweisen ist die Herleitung der Ortsdefinitionen der Kegelschnitte (aus denen sich dann deren Gleichungen ableiten lassen) mithilfe der DANDELINSchen Kugeln, siehe z. B. [235], S. 16ff. Eine vektorielle Herleitung der allgemeinen Scheitelgleichung der regulären Kegelschnitte enthält u. a. [224], S. 242f. Herleitungen der Gleichungen für Ellipsen, Parabeln und Hyperbeln, die ohne Vektoren auskommen und vor allem die Strahlensätze nutzen, enthält [97], S. 172f.

$$\varepsilon_1: z = 1,$$

$$\varepsilon_2: y = 1 \quad \text{und}$$

$$\varepsilon_3: y = z - 1$$

gegeben sind, leicht ermitteln:

$$(1) K \cap \varepsilon_1: x^2 + y^2 = 1, \quad (2) K \cap \varepsilon_2: -x^2 + z^2 = 1, \quad (3) K \cap \varepsilon_3: x^2 = 2z - 1 \quad \text{bzw.} \\ x^2 = 2y + 1.$$

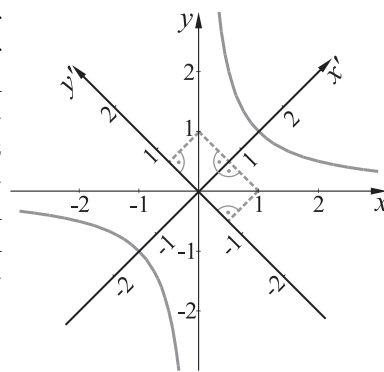
Dieses Vorgehen ist zwar nicht über Spezialfälle hinaus verallgemeinerbar, nimmt aber wenig Zeit in Anspruch und kann von den Schülern weitgehend selbstständig vollzogen werden. Es ermöglicht somit zumindest die ansatzweise Untersuchung der Kegelschnitte, die ansonsten kaum noch im Unterricht auftreten.<sup>95</sup> Nach der Herleitung der Gleichungen lassen sich diese durch das Einfügen verschiedener Koeffizienten variieren und auf diese Weise Verallgemeinerungen anstellen. Der Nachteil der beschriebenen einfachen Herleitung, dass sich keine Gleichung einer „echten“ Ellipse, sondern nur die eines Kreises ergibt, ist damit nachträglich kompensierbar. Die Schüler können die durch die drei Gleichungstypen mit unterschiedlichen Koeffizienten beschriebenen Kurven mithilfe eines Funktionenplotters bzw. eines CAS veranschaulichen und dabei feststellen, dass sich dadurch Ellipsen, Hyperbeln und Parabeln beliebiger Form generieren lassen.<sup>96</sup> Weitere zweidimensionale Überlegungen zu den Kegelschnitten, die u. a. ihre Eigenschaften als Ortslinien und im Zusammenhang damit die Brennpunkte betreffen, lassen sich anschließen. Interessant ist jedoch auch die Betrachtung analoger dreidimensionaler Objekte, auf die im Folgenden eingegangen wird.

Parabeln und Hyperbeln kennen die Schüler bereits aus dem MU der SI als Graphen elementarer Funktionen. Während bei der Parabel der Bezug zu der o. a. Gleichung (3) der als Kegelschnitt entstandenen Parabel leicht herzustellen ist, liegt dieser Bezug bei der Hyperbelgleichung (2) und einer Funktionsgleichung der Form  $f(x) = \frac{a}{x}$  nicht ohne weiteres auf der Hand. Die Schüler müssen dabei zunächst erkennen, dass sich die Lage der durch (2) beschriebenen Hyperbel in dem verwendeten Koordinatensystem und die der Funktionsgraphen um  $45^\circ$  unterscheiden, und dann eine Koordinatentransformation durchführen. Werden in der Gleichung  $y = \frac{1}{x}$  die Variablen mittels

$$x = \frac{1}{2}\sqrt{2}x' - \frac{1}{2}\sqrt{2}y', \quad y = \frac{1}{2}\sqrt{2}x' + \frac{1}{2}\sqrt{2}y'$$

durch  $x'$  und  $y'$  substituiert (siehe dazu die Abb. rechts), so ergibt sich  $x'^2 - y'^2 = 2$ .

Allerdings sind derartige Überlegungen für viele Schüler recht schwer zu vollziehen, da sie mit Koordinatentransformationen i. Allg. nicht vertraut sind. Zumindest sollte m. E. jedoch die Vorgehensweise erläutert werden, da es ausgesprochen unbefriedigend ist, wenn die hier und die in der SI behandelten Hyperbeln ohne Bezug zueinander bleiben.



<sup>95</sup>Es besteht bei der hier skizzierten speziellen Herleitung die Differenzierungsmöglichkeit, dass besonders leistungsstarke Schüler eine allgemeine Herleitung führen.

<sup>96</sup>Die Schüler sollten dabei feststellen, dass die Vergrößerung eines Koeffizienten nicht zu einer Streckung (im Sinne von Vergrößerung), sondern zu einer Stauchung entlang der entsprechenden Koordinatenachse führt. Oft erwarten Schüler, dass z. B. das Ersetzen des Terms  $x^2$  durch  $2x^2$  in der Gleichung  $x^2 + y^2 = 1$  die zugehörige Ellipse in  $x$ -Richtung vergrößert. Bei Experimenten fällt auf, dass das „Gegenteil“ eintritt (eine Streckung mit dem Faktor  $\frac{1}{\sqrt{2}}$ , also eine Stauchung in  $x$ -Richtung). Diese Erkenntnis ist z. B. durch das Umstellen der Gleichung zu fundieren. Betrachtungen der Schnittpunkte mit den Koordinatenachsen können dazu dienen, die geometrische Bedeutung der Koeffizienten  $a$  und  $b$  herauszuarbeiten, wenn die Ellipsengleichung in der Form  $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$  geschrieben wird.

### 4.5.2 Flächen zweiter Ordnung

Die Darstellung interessanter Flächen mithilfe einer Grafiksoftware bzw. eines CAS gehört sicherlich zu den ästhetisch attraktivsten Möglichkeiten des Computereinsatzes im Mathematikunterricht. Allerdings ist es für das mathematische Verständnis von Flächen wenig hilfreich, Gleichungen besonders attraktiver Flächen einer Formelsammlung oder einem Lehrbuch zu entnehmen, in eine Software einzugeben und lediglich einzelne Koeffizienten zu verändern sowie die Oberflächenerscheinung zu verschönern.<sup>97</sup> Vielmehr sollten die Schüler Zusammenhänge zwischen der mathematischen Beschreibung von Flächen und ihrer geometrischen Form erkennen und auf dieser Basis bewusst (z. B. mit Koeffizienten) experimentieren. Grundlagen hierfür bilden

- Variationen bereits bekannter Flächen,
- Analogieüberlegungen zu Kurven in der Ebene sowie
- Betrachtungen von Schnitten mit Koordinaten- bzw. dazu parallelen Ebenen.

Besonders geeignet für die Behandlung in der Schule sind Flächen zweiter Ordnung und Graphen von Funktionen zweier Variablen. Auch an die Beschreibung einiger Flächen durch Parameterdarstellungen ist in besonders leistungsstarken Kursen zu denken.

Die Untersuchung von Flächen zweiter Ordnung erscheint vor allem dann sinnvoll, wenn die Schüler zuvor bereits Gleichungen der Form

$$x^2 + y^2 + z^2 = r^2 \quad \text{und} \quad x^2 + y^2 - z^2 = 0$$

für Kugeln und (spezielle) Kegel in Mittelpunktslage hergeleitet haben. Günstig ist es zudem, wenn (wie im vorangegangenen Abschnitt beschrieben) Kegelschnitte untersucht wurden und deren Gleichungen bekannt sind. Nach den auf S. 264 erwähnten Variationen der Gleichungen (1), (2) und (3) und der Betrachtung der entstehenden Kurven in der Ebene, sollte den Schülern deutlich geworden sein, dass sich Ellipsen, Hyperbeln und Parabeln durch Gleichungen der Form<sup>98</sup>

$$(1') \quad A \cdot x^2 + B \cdot y^2 = 1, \quad (2') \quad A \cdot x^2 - B \cdot y^2 = 1, \quad (3') \quad x^2 = A \cdot y \quad \text{mit } A, B > 0 \text{ bzw.} \\ (1'') \quad \frac{x^2}{a^2} + \frac{y^2}{b^2} = 1, \quad (2'') \quad \frac{x^2}{a^2} - \frac{y^2}{b^2} = 1, \quad (3'') \quad x^2 = 2p \cdot y \quad \text{mit } a, b, p \neq 0$$

darstellen lassen. Diese Gleichungen können als Anhaltspunkte dienen, die Kugelgleichung so zu variieren, dass Flächen entstehen, deren Schnitte mit den Koordinatenebenen unterschiedliche Kegelschnitte sind. Dazu ist zu überlegen, dass durch das Weglassen des zu einer Koordinate gehörenden Terms in der Gleichung einer Fläche die Schnittkurve mit einer Koordinatenebene beschrieben wird.<sup>99</sup> Umgekehrt lassen sich

<sup>97</sup>Natürlich ist es reizvoll, Flächen in einer Grafiksoftware mit Texturen zu versehen und erfahrungsgemäß experimentieren Schüler damit gern. Problematisch ist es jedoch, wenn dabei mathematische Überlegungen zum Zusammenhang von Formen und Gleichungen von Flächen vernachlässigt werden.

<sup>98</sup>Bei der Parabelgleichung ist noch herauszuarbeiten, dass sich durch das Weglassen des absoluten Gliedes in der Gleichung (3) auf S. 264 lediglich eine Verschiebung der Parabel ergibt. Entsprechende Überlegungen sollten die Schüler bereits in der SI geführt haben.

<sup>99</sup>Dabei kann an die in 4.1.2–4.1.4 beschriebenen Betrachtungen zu Schnitten von Kugeln mit Koordinatenebenen und den Gleichungen der dabei entstehenden Schnittkreise angeknüpft werden.

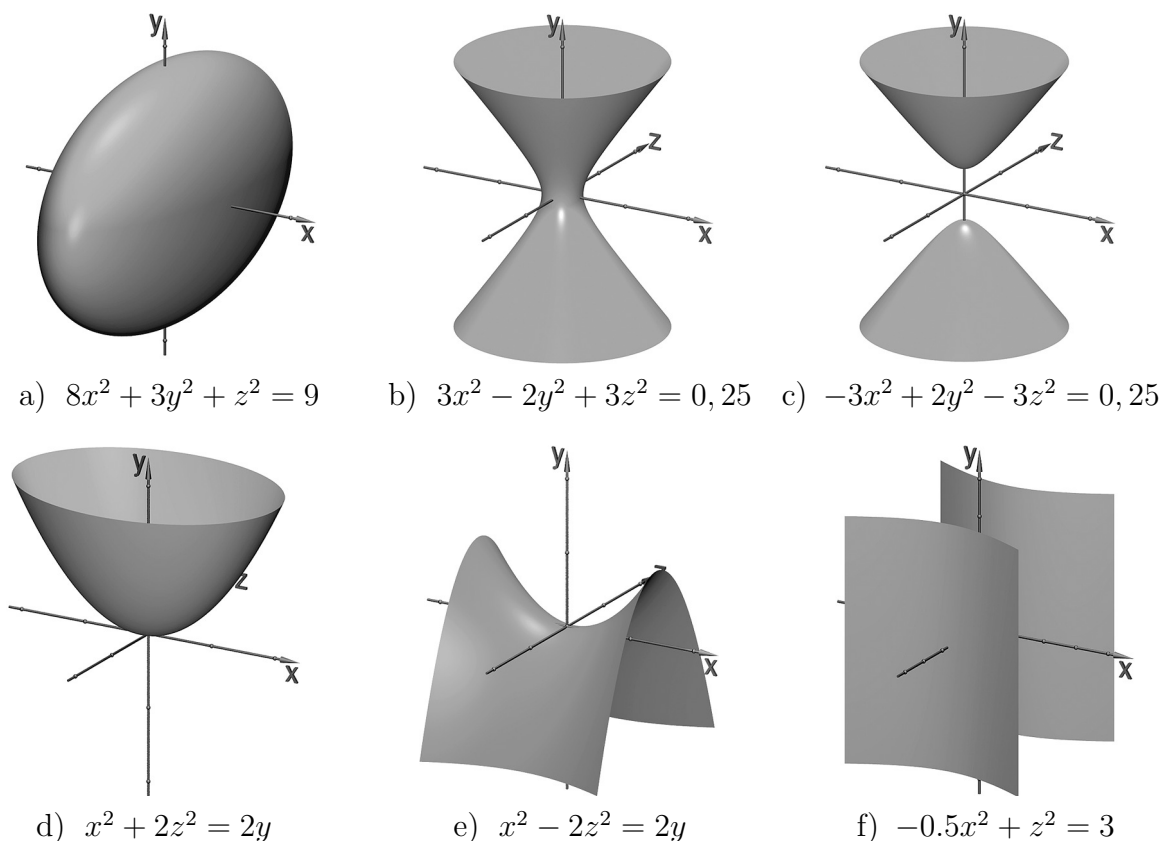


Abbildung 4.38: a) Ellipsoid, b) einschaliges Hyperboloid, c) zweischaliges Hyperboloid, d) elliptisches Paraboloid, e) hyperbolisches Paraboloid, f) hyperbolischer Zylinder

zu Gleichungen von Kurven Terme in der dritten Koordinate hinzufügen, woraus Gleichungen von Flächen entstehen.<sup>100</sup> Unter Berücksichtigung dieser Überlegungen können Schüler die Gestalt der durch verschiedene Gleichungen zweiter Ordnung beschriebenen Flächen mithilfe der verwendeten Software untersuchen<sup>101</sup> und dabei möglichst viele verschiedenartige Flächen finden.

Nach einer Phase des Experimentierens der Schüler sollten die Ergebnisse zusammengetragen und systematisiert werden, wobei m. E. keine vollständige Klassifikation aller Flächen zweiter Ordnung angestrebt werden muss, sondern vor allem Wert auf die geo-

<sup>100</sup>Hierbei bietet sich der Bezug zu Gleichungen von Geraden in der Ebene und Ebenen im Raum an.

<sup>101</sup>Kommt als Software POV-Ray zum Einsatz, so kann der `isosurface`-Befehl genutzt werden, der die Darstellung beliebiger durch implizite Gleichungen beschriebener Flächen ermöglicht, vgl. 2.3.5 und die Anleitung auf S. 337. Alternativ steht in POV-Ray für die Darstellung von Flächen mit Gleichungen der Form  $Ax^2 + By^2 + Cz^2 + Dxy + Exz + Fyz + Gx + Hy + Iz + J = 0$  auch der Befehl `quadric` zur Verfügung, der Grafiken wesentlich schneller berechnet. Allerdings ist es m. E. aus Gründen der Universalität dennoch sinnvoller, die `isosurface`-Anweisung zu verwenden. Bei Nutzung des CAS MuPAD können Flächen zweiter Ordnung durch `plot::Implicit3d` dargestellt werden, siehe S. 91. Beispieldateien zur Darstellung von Flächen zweiter Ordnung mithilfe von POV-Ray und MuPAD enthält der Ordner 4-5 (siehe S. 365f.).



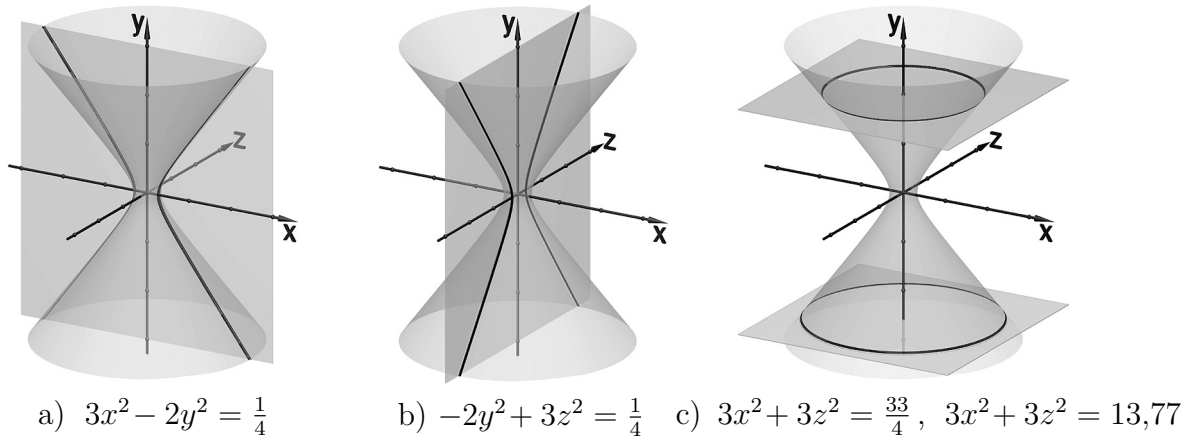


Abbildung 4.39: Schnitte des durch die Gleichung  $3x^2 - 2y^2 + 3z^2 = \frac{1}{4}$  beschriebenen Hyperboloids mit a) der  $x$ - $y$ -Ebene, b) der  $y$ - $z$ -Ebene, c) den Ebenen mit  $y = 2$  und  $y = -2$ ; Gleichungen der Schnittkurven

metrisch interessantesten Flächen, also Ellipsoide, ein- und zweischalige Hyperboloide sowie elliptische und hyperbolische Paraboloid (siehe Abb. 4.38) zu legen ist.<sup>102</sup>

Eine anschauliche „Klassifikation“ der von den Schülern gefundenen Flächen zweiter Ordnung ist durch die Betrachtung der Schnittkurven mit den Koordinatenebenen bzw. dazu parallelen Ebenen (wenn mit den entsprechenden Koordinatenebenen keine Schnittkurven existieren) möglich. Die Schnittkurven sollten sowohl visualisiert als auch anhand des Bezugs zu den Gleichungen, die durch Einsetzen von Null bzw. einer Konstanten für eine Koordinate entstehen, plausibel gemacht werden (siehe Abb. 4.39).

Fläche	Ellipsoid	Hyperboloid		Paraboloid	
		einschaliges Hyperboloid	zweischaliges Hyperboloid	elliptisches Paraboloid	hyperbolisches Paraboloid (Sattelfläche)
Gestalt der Gleichungen	$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1$ bzw. $Ax^2 + By^2 + Cz^2 = D$ ( $A, B, C, D > 0$ )	$\frac{x^2}{a^2} + \frac{y^2}{b^2} - \frac{z^2}{c^2} = 1$ bzw. $Ax^2 + By^2 - Cz^2 = D$ ( $A, B, C, D > 0$ )	$\frac{x^2}{a^2} - \frac{y^2}{b^2} - \frac{z^2}{c^2} = 1$ bzw. $Ax^2 - By^2 - Cz^2 = D$ ( $A, B, C, D > 0$ )	$\frac{x^2}{p} + \frac{y^2}{q} = 2z$ ( $p > 0, q > 0$ ) bzw. $Ax^2 + By^2 = Cz$ ( $A, B, C > 0$ )	$\frac{x^2}{p} - \frac{y^2}{q} = 2z$ ( $p > 0, q > 0$ ) bzw. $Ax^2 - By^2 = Cz$ ( $A, B, C > 0$ )
Schnittkurven mit KE	Ellipse, Ellipse, Ellipse	Ellipse, Hyperbel, Hyperbel	Ellipse,* Hyperbel, Hyperbel	Parabel, Parabel, Ellipse*	Parabel, Parabel, Hyperbel*
* Diese Kurven entstehen beim Schnitt der Flächen mit zu Koordinatenebenen parallelen Ebenen.					

<sup>102</sup>Neben diesen Flächen können auch Kegel, elliptische, parabolische und hyperbolische Zylinder sowie Punkte, Geraden, Ebenen und Paare paralleler Ebenen entstehen. Eine vollständige Untersuchung aller dieser Spezialfälle würde recht viel Zeit beanspruchen, weshalb es sinnvoll erscheint, dass die Schüler vor allem die „interessanteren“ Flächen suchen.

Zu den in der Übersicht auf S. 267 aufgeführten Gleichungen<sup>103</sup> ist anzumerken, dass Schüler eher mit Gleichungen der Form  $Ax^2 + By^2 + Cz^2 = D$  als mit der Schreibweise  $\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1$  arbeiten werden,<sup>104</sup> da dies bei der Arbeit mit einer Software näher liegt. Wie bereits zur Behandlung der Kegelschnitte angemerkt wurde, sollten sie jedoch den Zusammenhang zwischen den Koeffizienten und den Ausdehnungen der entsprechenden Flächen untersuchen, was bei der Darstellung verschiedener Flächen in einem Computerprogramm nahezu zwangsläufig erfolgt. Eventuell kann die geometrische Bedeutung der Koeffizienten  $a$ ,  $b$  und  $c$  zumindest für Ellipsoide herausgearbeitet werden.<sup>105</sup>

Im Zusammenhang mit der beschriebenen Untersuchung von Flächen zweiter Ordnung bzw. im Anschluss daran bietet es sich an, noch einige weitere Aspekte aufzugreifen:

- Durch die Anwendung ungleichmäßiger Streckungen (z. B. mithilfe der Anweisung `scale < $s_x, s_y, s_z$ >`) auf Kugeln können die Schüler ebenfalls *Ellipsoide* erzeugen und diese als (affine) *Bilder von Kugeln* betrachten.<sup>106</sup>
- Es lässt sich die Frage untersuchen, welche speziellen Flächen zweiter Ordnung *Rotationsachsen* besitzen und wie sich diese Eigenschaft in den Gleichungen der betreffenden Flächen niederschlägt.
- Mithilfe eines CAS oder einer Grafiksoftware ist es für die Schüler möglich, visuell zu untersuchen, durch welche Flächen zweiter Ordnung Geraden verlaufen (und somit, bei welchen dieser Flächen es sich um *Regelflächen* handelt).<sup>107</sup>

Weitere äußerst interessante Aspekte und Eigenschaften von Flächen 2. Ordnung sind in [93], S. 162ff. auf recht elementare Weise beschrieben.

<sup>103</sup>Es erfolgte eine Beschränkung auf Gleichungen in Mittelpunktslage. Die Betrachtung allgemeinerer Gleichungen ist m. E. nicht sinnvoll, da aus diesen oft nicht ohne weiteres ersichtlich ist, welche Flächen sie beschreiben. Die Behandlung der Hauptachsentransformationen dürfte in der Schule aufgrund des Schwierigkeitsgrades und der benötigten Voraussetzungen nicht in Frage kommen.

<sup>104</sup>Die auf S. 267 aufgeführten Gleichungen stehen nur exemplarisch für viele mögliche Schreibweisen und andere Zuordnungen von Vorzeichen zu Koordinaten – entscheidend für die Art der durch die Gleichungen beschriebenen Flächen ist ja die *Anzahl* der auftretenden negativen Vorzeichen.

<sup>105</sup>Dazu bedarf es jedoch geeigneter Hinweise des Lehrers, da es bei der experimentellen Untersuchung von Flächen nicht auf der Hand liegt, die absoluten Glieder zu normieren und die Wurzeln der Reziproken der entstehenden Koeffizienten zu bilden, Schüler dies also kaum von selbst tun werden.

<sup>106</sup>Analoge Überlegungen zu Kreisen und Ellipsen bieten sich bereits bei der Behandlung der Kegelschnitte an. Selbst wenn Kegelschnitte und Flächen 2. Ordnung ansonsten nicht behandelt werden, stellt die Erzeugung von Ellipsen und Ellipsoiden aus Kreisen bzw. Kugeln mithilfe von Streckungen eine recht einfache Möglichkeit dar, zumindest diese speziellen Kurven bzw. Flächen zu betrachten.

<sup>107</sup>Während diese Eigenschaft bei Kegeln, Ebenen und den verschiedenen Arten von Zylindern auf der Hand liegt, erfordert es mehr Vorstellung bzw. bedarf es sinnvoller Visualisierungen (z. B. der Schnitte mit geeigneten Ebenen), um zu erkennen, dass einschalige Hyperboloide und hyperbolische Paraboloiden Regelflächen sind. Hierzu bietet sich auch der Bau realer Modelle an (siehe z. B. [93], S. 169). Einen Weg, für Flächen mithilfe eines CAS analytisch zu entscheiden, ob es sich um Regelflächen handelt, beschrieb TIETZE in [247], S. 274f. Derartige Überlegungen dürften allerdings (wie TIETZE auch anmerkte) besonders begabten Schülern vorbehalten sein.

### 4.5.3 Untersuchung von Flächen als Graphen von Funktionen zweier Variablen

In der Untersuchung von Funktionsgraphen zweier Variablen dürfte sicherlich der einfachste Zugang zur Behandlung von Flächen bestehen. Dabei können Schüler zunächst mit der Eingabe einfacher Funktionsterme in die verwendete Software<sup>108</sup> beginnen und die Gestalt der dadurch erzeugten Funktionsgraphen betrachten. Bereits anhand der recht einfachen Funktionsgleichung  $f(x, y) = x \cdot y$  bzw.  $f(x, z) = x \cdot z$  (siehe Abb. 4.40 a) sind einige interessante Überlegungen möglich, wobei wiederum (sowohl visuell als auch anhand der Funktionsgleichung vorgenommene) Betrachtungen der Schnitte mit bestimmten Ebenen entscheidend für das Verständnis des Zusammenhangs zwischen der mathematischen Beschreibung und der geometrischen Form der Fläche sind. Während die Schnitte mit den Koordinatenebenen bei dem betrachteten Beispiel wenig überraschend die Koordinatenachsen sind, können die Schüler erkennen, dass die Schnittkurve der Fläche mit der durch die Gleichung  $x = z$  gegebenen Ebene (Abb. 4.40 d) eine parabelförmige Gestalt hat. Dies sollte zu der Überlegung führen, dass der Funktionsterm  $f(x, z) = x \cdot z$  für  $x = z$  zu  $f(x) = x^2$  wird.<sup>109</sup>

Anschließend können sich Aufgaben, bei denen Schüler für gewünschte Formen von Flächen geeignete Funktionsgleichungen finden und ihre Überlegungen dann durch Dar-

<sup>108</sup>Die Veranschaulichung von Funktionsgraphen zu Funktionsgleichungen der Form  $z = f(x, y)$  ist mit praktisch jedem CAS möglich. Bei Verwendung von POV-Ray kann hierfür der `parametric`-Befehl (siehe die Kurzanleitung auf S. 337f.) genutzt werden, indem statt eines Funktionsterms  $f(x, y)$  die Parameterdarstellung  $x = u$ ,  $y = v$ ,  $z = f(u, v)$  eingegeben wird. Aufgrund der Ausrichtung des Koordinatensystems in POV-Ray ist es oft sinnvoller, Funktionsgleichungen der Form  $y = f(x, z)$  und zugehörige Parameterdarstellungen  $x = u$ ,  $y = f(u, v)$ ,  $z = v$  zu verwenden. Der Ordner 4-5 enthält POV-Ray- und MuPAD-Dateien zu allen in diesem Abschnitt herangezogenen Beispielen (siehe S. 365f.).

<sup>109</sup>Allerdings greift diese Schlussweise etwas kurz und könnte den Eindruck erwecken, dass die Schnittkurve zu einer Normalparabel kongruent ist. Exakt wäre es, eine neue Variable  $\xi = \frac{\sqrt{2}}{2}x + \frac{\sqrt{2}}{2}z$  einzuführen (vgl. auch S. 264), über der sich für  $x = z$  eine quadratische Funktion mit der Funktionsgleichung  $f(\xi) = \frac{1}{2}\xi^2$  ergibt. Allerdings sind derartige Substitutionen für die Mehrzahl der Schüler recht schwer nachvollziehbar. Die geschilderte Überlegung ist demgegenüber sehr einfach anzustellen und hilft dennoch, die Form der Fläche zu erfassen und den Bezug zu der Funktionsgleichung zu verstehen.

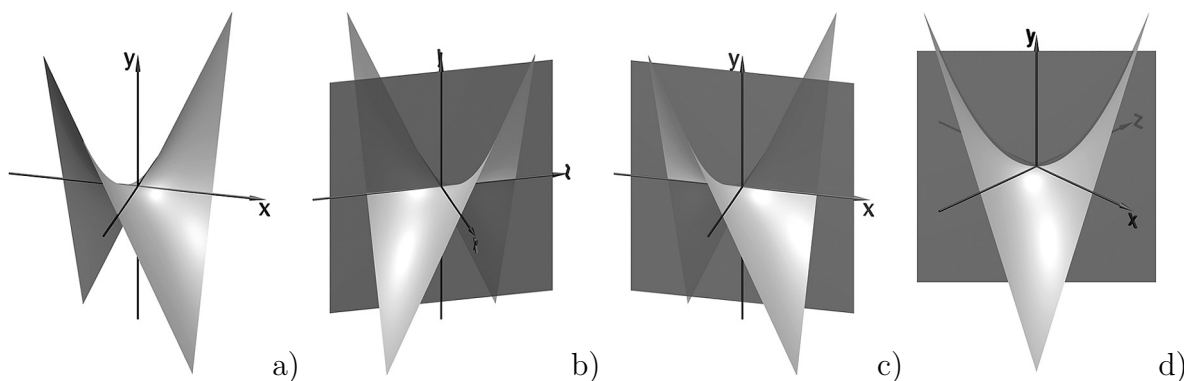


Abbildung 4.40: Durch die Gleichung  $y = x \cdot z$  beschriebene Fläche und Schnitte dieser Fläche mit der  $y$ - $z$ -Ebene, der  $x$ - $y$ -Ebene sowie der Ebene mit  $x = z$

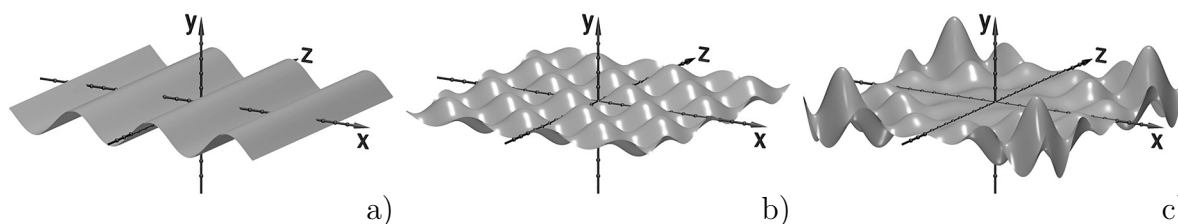


Abbildung 4.41: Graphen von Funktionen zweier Variablen

a)  $y = \sin x$     b)  $y = \sin x \cdot \sin z$     c)  $y = \frac{x \cdot y}{20} \cdot \sin x \cdot \sin z$  ;  $x, z \in [-10; 10]$

stellung der Flächen mithilfe der Software überprüfen. Dabei sollten zunächst relativ einfache Formen beschrieben und durch Variationen kompliziertere Gebilde modelliert werden. Die folgende *Sequenz von Aufgaben* soll ein solches Vorgehen verdeutlichen.

- Betrachten Sie die Abbildung 4.41 a). Überlegen Sie, durch welche Funktionsgleichungen die Kurven beschrieben werden können, die als Schnitte der dargestellten Fläche mit der  $x$ - $y$ - und der  $z$ - $y$ -Ebene entstehen. Finden Sie dann eine Gleichung für die Beschreibung der Fläche und überprüfen Sie, ob Sie dadurch tatsächlich eine Fläche der dargestellten Form erhalten.
- Betrachten Sie die Abbildung 4.41 b) und überlegen Sie, durch welche Gleichung diese „Wellenoberfläche“ beschrieben werden kann. Überprüfen Sie Ihre Überlegungen durch die Darstellung der Fläche mithilfe des Computers.
- Verändern Sie Ihre „Wellenoberfläche“ aus der vorangegangenen Aufgabe so, dass in der Mitte ein nahezu „ruhiger Seegang“ herrscht und die Wellen nach außen hin höher werden.

Eine mögliche Lösung der letzten Aufgabe zeigt Abb. 4.41 c); diese Abbildung (und auch die zugehörige Gleichung) sollten die Schüler jedoch bei der vorgeschlagenen Aufgabensequenz nicht vorgelegt bekommen, sondern selbst entwickeln. In der Herangehensweise analoge Modellierungen von Formen sind natürlich auch anhand völlig anderer Beispiele unter Verwendung unterschiedlicher elementarer Funktionen durchführbar.

#### 4.5.4 Zugänge zu Parameterdarstellungen von Flächen

Falls sich Schüler (z. B. im Zusammenhang mit Bewegungsbahnen, vgl. Abschnitt 4.3.1) bereits mit Parameterdarstellungen von Kreisen und evtl. daraus abgeleiteten Kurven beschäftigt haben, so ist in Erwägung zu ziehen, durch einige zusätzliche Überlegungen auch Flächen zu untersuchen, die sich nicht als Funktionsgraphen darstellen lassen, sondern durch Parameterdarstellungen beschrieben werden müssen.<sup>110</sup> Als nahe liegendes und variationsfähiges Beispiel einer ersten durch eine Parameterdarstellung darzustellenden Fläche bietet sich der *Zylinder* an. Ausgehend von der Parameterdarstellung

$$x(\alpha) = r \cdot \cos \alpha, \quad z(\alpha) = r \cdot \sin \alpha; \quad \alpha \in [0; 2\pi)$$

<sup>110</sup>Dabei kann auch die Analogie zu Parameterdarstellungen von Geraden (als einparametrische Gebilde) und Ebenen (mit zwei Parametern) herangezogen werden und auf analoge Weise von Parameterdarstellungen der in Abschnitt 4.3.1 betrachteten Kurven zu Flächen übergegangen werden.

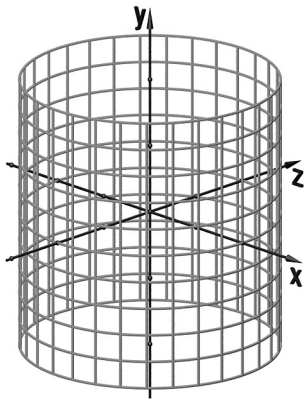


Abbildung 4.42: Parameterlinien eines Zylinders

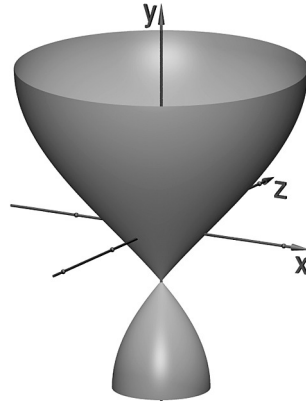


Abbildung 4.43: Rotationsfläche, die durch Variation der Parameterdarstellung eines Zylinders entsteht

eines Kreises in der  $x$ - $z$ -Ebene<sup>111</sup> (siehe S. 250) ist anschaulich leicht zu überlegen, dass durch Zuordnung der dritten (in der Gleichung des Kreises nicht auftretenden) Koordinate zu einem zweiten Parameter die Menge aller durch Punkte des Kreises verlaufenden Geraden entsteht,<sup>112</sup> die auf der  $x$ - $z$ -Ebene senkrecht stehen (also parallel zur  $y$ -Achse sind). Es ergibt sich somit eine Zylinderfläche mit der Parameterdarstellung

$$x(u, v) = r \cdot \cos u, \quad y(u, v) = v, \quad z(u, v) = r \cdot \sin u \quad \text{mit } u \in [0; 2\pi), v \in [v_1; v_2].$$

Diese kann gleichzeitig als Vereinigungsmenge von Kreisen in zur  $x$ - $z$ -Ebene parallelen Ebenen aufgefasst werden, deren Mittelpunkte auf der  $y$ -Achse liegen (siehe Abb. 4.42). Eine mögliche Variation der Zylinderfläche besteht darin, den Radius  $r$  dieser Kreise nicht konstant zu belassen, sondern in Abhängigkeit von  $v$  und somit von der „Höhe“ der Kreise (bei einer Anordnung des Zylinders wie in Abb. 4.42) auszudrücken. Die Mantellinien des Zylinders „mutieren“ dabei zu Graphen von Funktionen  $r = f(v) = f(y)$ ; es entstehen von diesen Funktionsgraphen erzeugte Rotationsflächen. Wird diese anschauliche Interpretation einer speziellen Klasse von Parameterdarstellungen für die Schüler verständlich, so können sie durch die Verwendung geeigneter Funktionsterme  $r(v)$  eine Vielzahl von Rotationsflächen erzeugen. Als ein Beispiel zeigt Abb. 4.43 die durch den Funktionsgraphen mit der Gleichung  $r(v) = 0,5 + \sin v$  erzeugte Rotationsfläche mit der Parameterdarstellung

$$\begin{aligned} x(u, v) &= (0,5 + \sin v) \cdot \cos(u) \\ y(u, v) &= v \\ z(u, v) &= (0,5 + \sin v) \cdot \sin(u) \end{aligned} \quad \text{mit } u \in [0; 2\pi), v \in \left[-\frac{\pi}{2}; \frac{\pi}{2}\right].$$

Ein weiterer Ansatz zur Modellierung von Rotationsflächen besteht in der Betrachtung konzentrischer Kreise in einer Ebene (z. B. der  $x$ - $z$ -Ebene). Diese Ebene lässt sich als Vereinigungsmenge aller in ihr liegenden Kreise mit dem Mittelpunkt im Koordinatenursprung auffassen und durch die Parameterdarstellung

$$x(u, v) = v \cdot \cos u, \quad y(u, v) = 0, \quad z(u, v) = v \cdot \sin u; \quad u \in [0; 2\pi), v \in [0; \infty)$$

<sup>111</sup>Natürlich kann auch von einem Kreis in einer anderen Koordinatenebene ausgegangen werden.

<sup>112</sup>Wird für den zweiten Parameter ein endliches Intervall gewählt, so handelt es sich nur um Strecken. Für die Erkennbarkeit der Form der entstehenden Figur ist dies sinnvoller.

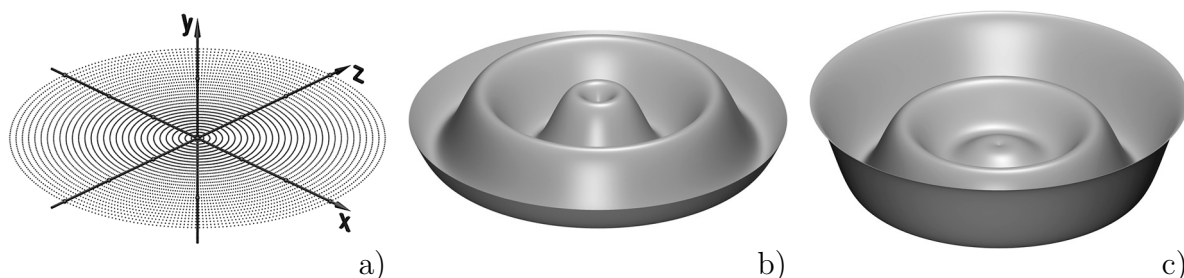


Abbildung 4.44: Ebene als Menge konzentrischer Kreise (a) und durch Variation der Parameterdarstellung daraus abgeleitete Rotationsflächen (b, c)

beschreiben (siehe Abb. 4.44 a). Der Parameter  $v$  gibt dabei die Radien der Kreise an. Wird  $y$  als Funktion von  $v$  (mit  $v > 0$ ), also der Entfernung vom Koordinatenursprung aufgefasst, so ergeben sich Rotationsflächen, die durch Drehung des Funktionsgraphen mit der Gleichung  $y = f(v)$  um die  $y$ -Achse entstehen. Abb. 4.44 zeigt zwei Beispiele, wobei die Funktionsgleichungen b)  $y(v) = 2 \cdot \sin v$  und c)  $y(v) = \frac{v}{4} \cdot \sin v$  genutzt wurden. Parameterdarstellungen dieser Rotationsflächen sind somit

b)  $x(u,v) = v \cdot \cos u$ ,  $y(u,v) = 2 \cdot \sin v$ ,  $z(u,v) = v \cdot \sin u$ ;  $u \in [0; 2\pi)$ ,  $v \in [0; 4\pi]$  bzw.

c)  $x(u,v) = v \cdot \cos u$ ,  $y(u,v) = \frac{v}{4} \cdot \sin v$ ,  $z(u,v) = v \cdot \sin u$ ;  $u \in [0; 2\pi)$ ,  $v \in [0; 4\pi]$ .

Die Beispiele zeigen, dass sich durch Rotationsflächen vielfältige Formen modellieren lassen und gleichzeitig anschauliche Zugänge zu ihrer mathematischen Beschreibung erarbeitet werden können. Dabei sind die hier beschriebenen Zugänge, Rotationsflächen als spezielle Flächen zweiter Ordnung, als Variationen von Zylindern sowie von Mengen konzentrischer Kreise einer Ebene zu betrachten, nicht die einzig möglichen.<sup>113</sup> Denkbar wäre es auch, Beschreibungen durch implizite Gleichungen vorzunehmen, wobei nicht unbedingt quadratische (bzw. polynomiale) Terme auftreten müssen, ein Beispiel enthält S. 91.<sup>114</sup> Variationen der beschriebenen Rotationsflächen ergeben sich durch Multiplikation der Terme  $x(u,v)$  und  $z(u,v)$  mit unterschiedlichen Koeffizienten bzw. (mit gleichem Ergebnis) die Anwendung ungleichmäßiger Streckungen auf die Flächen.

Aus der Vielfalt der möglichen, allerdings bereits recht anspruchsvollen, Ansätze zur mathematischen Beschreibung interessanter Flächen sei im Folgenden noch kurz auf einen Zugang zur *Parameterdarstellung der konischen Spiralfäche*<sup>115</sup> eingegangen. Voraussetzung dafür ist, dass die Schüler mit der entsprechenden Kurve (der konischen Spirale) bereits gearbeitet haben, z. B. im Zusammenhang mit Bahnkurven, siehe Abschnitt 4.3.1. Als Ausgangspunkt der Überlegungen zur konischen Spiralfäche wird dann diese Kurve mit einer Parameterdarstellung wie z. B.

<sup>113</sup>Von Flächen zweiter Ordnung ausgehende didaktische Überlegungen zur Behandlung von Rotationsflächen finden sich u. a. in [247], S. 262ff.

<sup>114</sup>Entscheidend für das Verständnis des Zusammenhangs zwischen den Gleichungen und den geometrischen Formen der Flächen sind bei einem derartigen Zugang wiederum Überlegungen zu den Schnitten der beschriebenen Flächen mit den Koordinatenebenen.

<sup>115</sup>Als „konische Spiralen“ werden sowohl Kurven als auch Flächen bezeichnet. Um Verwechslungen zu vermeiden, wird im Folgenden die (unübliche) Bezeichnung „konische Spiralfäche“ verwendet.

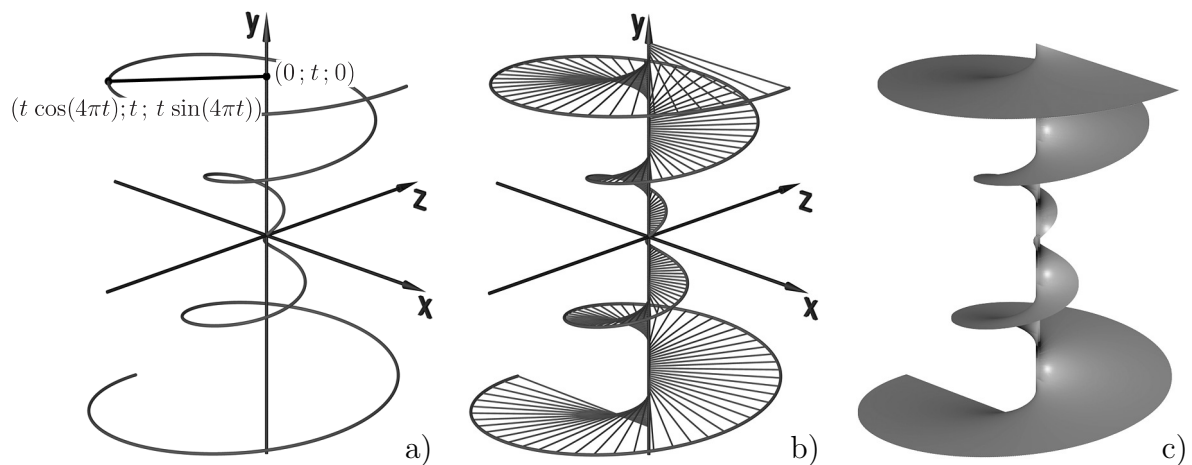


Abbildung 4.45: Aufbau einer konischen Spiralfäche aus den Loten von den Punkten der konischen Spirale auf eine Koordinatenachse (siehe auch Anhang B, S. 351)

$$x(t) = t \cdot \cos(4\pi t), \quad y(t) = t, \quad z(t) = t \cdot \sin(4\pi t); \quad t \in [-1; 1]^{116}$$

herangezogen und die Überlegung angeregt, wie Lote von den Punkten der Kurve auf die  $y$ -Achse beschrieben werden können. Dabei handelt es sich um die Verbindungsstrecken zwischen den durch die Koordinaten  $(t \cdot \cos(4\pi t); t; t \cdot \sin(4\pi t))$  gegebenen Punkten der Kurve und den Punkten  $(0; t; 0)$  auf der  $y$ -Achse (siehe Abb. 4.45 a). Diese Strecken lassen sich durch Parameterdarstellungen der Form

$$\begin{aligned} x(s) &= 0 + s \cdot (t \cdot \cos(4\pi t) - 0) \\ y(s) &= t + s \cdot (t - t) \\ z(s) &= 0 + s \cdot (t \cdot \sin(4\pi t) - 0) \end{aligned} \quad ; \quad s \in [0; 1]$$

mit konstantem  $t$  darstellen. Die Vereinigungsmenge aller dieser Strecken bildet die konische Spiralfäche (Abb. 4.45 b), deren Parameterdarstellung sich aus den Parameterdarstellungen der Strecken ergibt, wobei die darin als konstant angenommene Größe  $t$  als zweiter Parameter auftritt:

$$x(s, t) = s \cdot t \cdot \cos(4\pi t), \quad y(s, t) = t, \quad z(s, t) = s \cdot t \cdot \sin(4\pi t); \quad s \in [0; 1], \quad t \in [-1; 1].$$

Sicherlich ist die skizzierte Parametrisierung der konischen Spiralfäche ein sehr anspruchsvolles Beispiel, das zudem einige Voraussetzungen erfordert: Die Schüler müssen zuvor Parameterdarstellungen von Kreisen entwickelt und variiert haben, so dass ihnen die Parameterdarstellung der konischen Spirale (als Kurve) vertraut ist. Zudem müssen sie Parameterdarstellungen von Geraden bzw. Strecken flexibel anwenden können. Diese Voraussetzungen dürften bei vielen Schülern nicht in dem notwendigen Umfang erreicht werden. In sehr leistungsstarken Kursen bzw. für besonders interessierte Schüler ist jedoch die Untersuchung der konischen Spiralfäche m. E. eine interessante Möglichkeit der Behandlung eines attraktiven Objekts.

<sup>116</sup>Im Sinne der Einfachheit und um damit zu verdeutlichen, dass es zumindest für leistungsstarke Schüler möglich ist, die beschriebenen Überlegungen zu vollziehen, werden diese hier anhand eines Beispiels dargestellt. Dessen Verallgemeinerung ist jedoch leicht möglich.

### 4.5.5 Zusammenfassung und Einordnung

Das wichtigste Ziel der vorgeschlagenen Untersuchungen von Flächen zweiter Ordnung wie auch von anderen Flächen des Raumes besteht darin, dass Schüler exemplarisch Zusammenhänge zwischen mathematischen Beschreibungen und geometrischen Objekten anhand interessanter Körper und Flächen durch eigene Experimente erkennen und dadurch Fähigkeiten der Modellierung entwickeln. Die computerunterstützte Visualisierung ist dabei eine wertvolle Hilfe, soll aber den Schülern vor allem dabei helfen, eigene anschauliche Vorstellungen herauszubilden und die Fähigkeit auszuprägen, Visualisierungen durch Gleichungen beschriebener Objekte auch „im Kopf“ herzustellen. Umgekehrt sollen sie überlegen, wie Gleichungsterme variiert werden müssen, um gewünschte Formen darzustellen. Die Betrachtung von Schnittfiguren sowie die damit verbundene Transformation ebener in räumliche Sachverhalte und umgekehrt erweisen sich dabei als zentrale Strategien.

Die beschriebenen Zugänge zu Kegelschnitten und Flächen zweiter Ordnung, zu Graphen von Funktionen zweier Variablen sowie zu Parameterflächen sind unabhängig und bauen nicht aufeinander auf.<sup>117</sup> Aus Zeitgründen wird es ohnehin nicht möglich sein, alle in diesem Abschnitt beschriebenen Betrachtungen durchzuführen. Die geringsten Voraussetzungen sind für die *Betrachtungen von Funktionsgraphen zweier Variablen* erforderlich. Hierzu kann an den Analysislehrgang bzw. sogar an die Behandlung der elementaren Funktionen in der SI angeknüpft werden. Für die Behandlung der *Kegelschnitte und der Flächen zweiter Ordnung* sollten die Schüler zuvor Kreise, Kugeln und Kegel untersucht und durch Gleichungen beschrieben haben, wie z. B. in Abschnitt 4.1 dargelegt wurde. Um *Flächen durch Parameterdarstellungen* zu beschreiben, sollten die Schüler dies vorher unbedingt für Kurven getan haben (siehe Abschnitt 4.3) – die hier vorgeschlagenen Betrachtungen allein durch Variationen der Parameterdarstellungen von Geraden und Ebenen zu führen, erscheint m. E. wenig aussichtsreich.

Fast alle geometrischen Überlegungen in den Abschnitten 4.1, 4.3, 4.5.1, 4.5.2 und 4.5.4 gingen von Kreisen und Kugeln aus. Durch Variationen daran lassen sich Kurven und Flächen modellieren, deren Formenreichtum m. E. der Formulierung ZEITLERS von der Geometrie als „Zaubergarten“ entspricht (siehe S. 21). Es ist deshalb in Erwägung zu ziehen, in einem geometrisch orientierten Lehrgang der analytischen Geometrie „*Kreise, Kugeln und ihre Verwandten*“ in den Mittelpunkt zu stellen. Dies würde allerdings die Behandlung von Elementen der linearen Algebra erschweren, die zur Dominanz der geometrisch vergleichsweise uninteressanten Geraden und Ebenen geführt hat.

Es handelt sich bei den hier vorgeschlagenen Überlegungen nicht um „neue“ mathematische Inhalte, die durch die Computergrafik entstanden sind, sondern um klassische Gegenstände der analytischen Geometrie – jedoch lassen die Visualisierungsmöglichkeiten, die der Computer bietet, gerade die Behandlung von Flächen erst realistisch erscheinen.

<sup>117</sup>Neben den beschriebenen Zugängen gibt es weitere Möglichkeiten, Klassen von Flächen im Unterricht zu thematisieren. So gingen SCHROTH und TIETZE in [247], S.271ff. z. B. auf Regelflächen ein.



## 4.6 Einbeziehung mathematischer Grundlagen der 3D-Computergrafik bei der Behandlung des Skalarproduktes und von Normalenvektoren

Aus Abschnitt 2.5 geht hervor, dass die Bestimmung von Winkeln bzw. Skalarprodukten sowie die Berechnung, Mittelung und Interpolation von Normalenvektoren zentrale Bestandteile der Bildberechnung in der 3D-Computergrafik sind. Ziel des vorliegenden Abschnitts ist es, dies für die Motivierung und Anwendung traditioneller Unterrichtsinhalte sowie für Modellbildungsprozesse (siehe S. 49f.) im Unterricht zu nutzen.

Bei den im Folgenden dargestellten Überlegungen kommen beide Aspekte (W) und (G) der Einbeziehung der 3D-Computergrafik (siehe Abschnitt 3.2.2, S. 189) zum Tragen. Als *Werkzeug* wird der Computer genutzt, um die Behandlung des Skalarproduktes und von Normalenvektoren visuell zu unterstützen. Vor allem aber erfolgt eine Thematisierung von Elementen der Wirkungsweise der fotorealistischen Computergrafik, die somit zu einem *Unterrichtsgegenstand* wird, der in engem Zusammenhang mit den genannten mathematischen Inhalten steht. Es werden Beispiele aufgezeigt, anhand derer Schüler durch die Erarbeitung und Nutzung von Beleuchtungsmodellen sowie die Berechnung von Normalenvektoren ausgewählte mathematische Grundlagen der Computergrafik bei der Erstellung eigener Grafiken selbst nachvollziehen und anwenden können.

### 4.6.1 Das Reflexionsgesetz als Ausgangspunkt der Behandlung des Skalarproduktes

Ein Einstieg in den Stoffabschnitt zum Skalarprodukt und seinen Anwendungen kann darin bestehen, die Wirkungsweise des Raytracing-Verfahrens z. B. in einem Schülervortrag vorzustellen.<sup>118</sup> Dies ist natürlich vor allem dann sinnvoll, wenn die Schüler vorher bereits mit einer Raytracing-Software wie POV-Ray gearbeitet haben und somit wissen, dass durch Raytracing fotorealistische Grafiken generiert werden können und sich dabei insbesondere Objekte in anderen Objekten spiegeln. Daher liegt es unmittelbar auf der Hand, dass bei der Generierung fotorealistischer Grafiken Reflexionen berechnet werden müssen. Bereits im Physikunterricht der SI haben die Schüler das Reflexionsgesetz in der folgenden bzw. einer ähnlichen Formulierung kennengelernt:

*Der Winkel  $\alpha$  zwischen dem einfallenden Lichtstrahl und dem Einfallslot (Einfallswinkel) und der Winkel  $\beta$  zwischen dem reflektierten Lichtstrahl und dem Einfallslot (Reflexionswinkel) sind maßgleich.*

Als Einfallslot wird die auf der spiegelnden Oberfläche im Auftreffpunkt des einfallenden Lichtstrahls errichtete Senkrechte bezeichnet. In einigen Schulbüchern findet sich dazu eine Anmerkung, dass die beiden Strahlen und das Einfallslot in einer Ebene liegen; ansonsten geht dies aus einer Skizze bzw. einem Versuchsaufbau hervor und wird als selbstverständlich betrachtet.

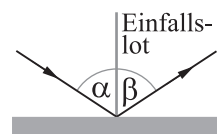


Abbildung 4.46: Reflexionsgesetz in einer Ebene

<sup>118</sup>Wie bereits erwähnt wurde, eignet sich dieses Verfahren aufgrund seiner prinzipiell recht einfachen und auf elementaren Gesetzen der geometrischen Optik beruhenden Funktionsweise besonders gut dafür, im Unterricht thematisiert zu werden.

Um das Reflexionsgesetz, welches im Physikunterricht der SI lediglich in der Ebene betrachtet wird, auf den dreidimensionalen Fall zu übertragen, ist es notwendig, eine *Entsprechung für das Einfallslot* zu finden, also Geraden bzw. Vektoren anzugeben, die auf gegebenen Flächen in den Lichteinfallspunkten senkrecht stehen.<sup>119</sup> Dass dabei Ebenennormalen von besonderer Bedeutung sind, ergibt sich aus der Tatsache, dass „reale“ Objekte in der 3D-Computergrafik meist durch Dreiecksnetze repräsentiert werden (vgl. Abschnitt 2.3.9).<sup>120</sup> Normalen von Facetten sind somit die Normalen der die entsprechenden Dreiecke enthaltenden Ebenen. Neben Normalen sind *Winkel zwischen Einfallsloten und Lichtstrahlen* zu bestimmen; auch dies wird anhand des Reflexionsgesetzes sofort deutlich.

Eine Einführung des Skalarproduktes mit der Bildberechnung in der Computergrafik (und damit dem Reflexionsgesetz) als Ausgangspunkt rückt also geometrische Überlegungen (Bestimmung von Winkeln und als Spezialfall Orthogonalitätsuntersuchungen) in den Vordergrund. Allerdings erscheint es wenig realistisch, dass eine nennenswerte Zahl von Schülern anhand der beschriebenen Überlegungen und geeigneter Aufgaben selbst zum Skalarprodukt bzw. zu Möglichkeiten der Berechnung des Winkels zweier Vektoren findet. Es wird also notwendig sein, den Schülern bei den theoretischen Überlegungen zum Skalarprodukt geeignete Ansätze vorzugeben.

Eine Einführung des Skalarproduktes, die sich unmittelbar an computergrafischen Anwendungen bzw. Experimenten orientiert, liegt nicht auf der Hand. Die hier dargestellte Motivierung durch Anwendungen in der Computergrafik ist somit gewissermaßen eine Motivierung „auf Vorrat“, die anhand erst „später“ (wenn das Skalarprodukt zur Verfügung steht) möglicher Anwendungen bei der Erstellung von Grafiken zum Tragen kommt. Wie bereits in den Abschnitten 1.4.6 und 3.2 erwähnt wurde, ist die motivierende Wirkung von Anwendungen mathematischer Inhalte, die durch die Schüler nicht aktiv (nach)vollzogen werden können, begrenzt – auch eine Motivierung hinsichtlich später möglicher Anwendungen ist somit nicht unproblematisch. Ein solches Vorgehen erscheint m. E. im Falle des Skalarproduktes und seiner Anwendungen, die für die Bildberechnung in der Computergrafik essenziell sind, dennoch sinnvoll, wenn die Schüler bereits zuvor bereits mit einer Grafiksoftware gearbeitet und fotorealistische Computergrafiken generiert haben. Als Ausblicke auf computergrafische Anwendungen des Skalarproduktes und von Normalenvektoren können dazu die (durch Schüler selbst vorzunehmende) Glättung von durch Dreiecksnetzen repräsentierten Objekten sowie ein Verständnis wichtiger, das Aussehen von Oberflächen bestimmender, Renderparameter genannt werden.

<sup>119</sup>Für Ebenen sind die Normalenvektoren natürlich unabhängig vom betrachteten Punkt. Für gekrümmte Flächen müsste der Exaktheit halber von Senkrechten zu den Tangentialebenen der Flächen in den betrachteten Punkten gesprochen werden. Bei Kugeloberflächen dürften die Schüler jedoch auch ohne ausführliche Beschäftigung mit den Tangentialebenen die Normalen erkennen.

<sup>120</sup>Die Repräsentation von Objekten durch Dreiecksnetze kann den Schülern mithilfe von Abbildungen wie auf S. 110 und S. 136 leicht verdeutlicht werden.

### 4.6.2 Zur Einführung des Skalarproduktes

Zum Skalarprodukt bestehen recht unterschiedliche Zugänge, wobei sich geometrische, arithmetische (oft außermathematisch motivierte), strukturorientierte sowie „gemischte“ Einführungen unterscheiden lassen.<sup>121</sup> Da bei einer Einbeziehung von Elementen der 3D-Computergrafik in den Unterricht geometrische Fragestellungen im Vordergrund stehen, gleichzeitig jedoch die analytische Geometrie koordinatenorientiert behandelt wird, erscheint ein „gemischter“ (arithmetisch-geometrischer) Zugang zum Skalarprodukt am geeignetsten. Dabei sollten folgende Bedingungen beachtet werden:

- Da die Berechnung von Winkeln das zentrale *Ziel* der Einführung des Skalarproduktes darstellt, ist eine *Definition* des Skalarproduktes zweier Vektoren unter Verwendung des Winkels zwischen diesen Vektoren wenig sinnvoll – eine *Einführung anhand der Koordinatendarstellung* erscheint insofern geeigneter.
- Eine unmotiviert Einführung der Koordinatendarstellung des Skalarproduktes (wie in [27]) oder eine durch nichtgeometrische Anwendungen motivierte Einführung (wie in [97]) erscheinen ungünstig. Die Schüler sollten zunächst erkennen, dass es im Hinblick auf die Beschreibung der Orthogonalität von Vektoren und die Berechnung von Winkeln *sinnvoll ist, den Term  $x_a x_b + y_a y_b + z_a z_b$  zu betrachten*.

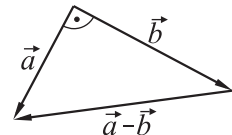
Wenn nach dem in Abschnitt 4.1.3 beschriebenen Vorgehen recht früh mithilfe des Satzes des Pythagoras eine Gleichung für den Abstand zweier Punkte des Raumes hergeleitet wird, kann darauf bei der Einführung des Skalarproduktes zurückgegriffen werden.

Für die „Länge“ (den Betrag) eines Vektors  $\vec{x} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$  ergibt sich daraus

$$|\vec{x}| = \sqrt{x^2 + y^2 + z^2} \quad \text{bzw.} \quad |\vec{x}|^2 = x \cdot x + y \cdot y + z \cdot z.$$

Daran anknüpfend können die Schüler überlegen, unter welcher Bedingung zwei Vektoren orthogonal (senkrecht zueinander) sind, wobei sich auch hierzu die Verwendung des Satzes des Pythagoras sowie seiner Umkehrung anbietet. Danach sind zwei Vektoren  $\vec{a}$  und  $\vec{b}$  genau dann orthogonal, falls

$$|\vec{a}|^2 + |\vec{b}|^2 = |\vec{a} - \vec{b}|^2$$



<sup>121</sup>Da eine Erläuterung der verschiedenen Zugänge erheblich den Rahmen dieser Arbeit überschreiten würde, sei hierzu u. a. auf [208] sowie [247], S. 185ff. verwiesen. In Schulbüchern wird das Skalarprodukt auf sehr unterschiedlichen Wegen eingeführt, wobei sich alle der genannten Zugänge finden lassen (axiomatische bzw. strukturorientierte Zugänge jedoch nicht mehr in neueren Lehrwerken). So enthält [97] eine arithmetische, anhand der Koordinaten zweier Vektoren vorgenommene Einführung (ausgehend von Stücklisten und Preisvektoren). Ebenfalls anhand der Koordinatendarstellung wird das Skalarprodukt in [27], [216], [217] und [224] definiert, wobei in [216] und [217] geometrische Vorüberlegungen angestellt werden, aus denen sich nach der Definition des Skalarproduktes recht schnell der Zusammenhang  $\langle \vec{a}, \vec{b} \rangle = |\vec{a}| \cdot |\vec{b}| \cdot \cos \angle(\vec{a}, \vec{b})$  ergibt. In [27], [97] und [224] erfolgt erst nach einer (sehr schnell gegebenen) Definition  $\langle \vec{a}, \vec{b} \rangle = x_a x_b + y_a y_b$  bzw.  $\langle \vec{a}, \vec{b} \rangle = x_a x_b + y_a y_b + z_a z_b$  eine Herleitung des Zusammenhangs zu dem Winkel zwischen beiden Vektoren. Geometrische Definitionen des Skalarproduktes finden sich u. a. in [8] und [223], wobei in [223] die Motivierung der Definitionsgleichung  $\langle \vec{a}, \vec{b} \rangle = |\vec{a}| \cdot |\vec{b}| \cdot \cos \angle(\vec{a}, \vec{b})$  physikalisch (unter Betrachtung der Arbeit an einer geneigten Ebene) erfolgt.

gilt. Diese Bedingung lässt sich unter Verwendung von Koordinaten auch in der Form

$$\begin{aligned} x_a^2 + y_a^2 + z_a^2 + x_b^2 + y_b^2 + z_b^2 &= (x_a - x_b)^2 + (y_a - y_b)^2 + (z_a - z_b)^2 \quad \text{bzw.} \\ 0 &= -2x_ax_b - 2y_ay_b - 2z_az_b, \quad \text{also} \\ x_ax_b + y_ay_b + z_az_b &= 0 \end{aligned} \quad (4.1)$$

schreiben. Damit ist durch recht elementare Überlegungen die Bedeutung des Terms  $x_ax_b + y_ay_b + z_az_b$  in den Mittelpunkt gerückt, der fortan als *Skalarprodukt*  $\langle \vec{a}, \vec{b} \rangle$  bzw.  $\vec{a} \cdot \vec{b}$  bezeichnet und weiter untersucht werden kann. Durch die vorherigen Überlegungen ist bereits bekannt, dass  $\langle \vec{a}, \vec{a} \rangle = |\vec{a}|^2$  ist und  $\langle \vec{a}, \vec{b} \rangle = 0$  genau dann gilt, wenn die Vektoren  $\vec{a}$  und  $\vec{b}$  orthogonal sind. Recht leicht lässt sich auch herleiten, dass für kollineare Vektoren  $\langle \vec{a}, \vec{b} \rangle = |\vec{a}| \cdot |\vec{b}|$  gilt. Damit liegt die Vermutung nahe, dass ein Zusammenhang zwischen dem Skalarprodukt und dem Winkel zwischen zwei Vektoren besteht. Um diesen Zusammenhang näher zu untersuchen, können Schüler Paare von Vektoren in der Ebene zeichnen, ihre Beträge und Skalarprodukte berechnen sowie die Winkel messen (siehe z. B. [224], S. 168, Aufgabe 9) bzw. im Raum analoge Betrachtungen mithilfe des Computers anstellen. Eine entsprechende Aufgabe kann folgendermaßen lauten:

A. Gegeben sind zwei Vektoren  $\vec{a}$  und  $\vec{b}$ .

$$\text{a) } \vec{a} = \begin{pmatrix} -2 \\ 5 \\ 3 \end{pmatrix}, \vec{b} = \begin{pmatrix} 2 \\ 2 \\ -1 \end{pmatrix} \quad \text{b) } \vec{a} = \begin{pmatrix} 6 \\ 6 \\ -3 \end{pmatrix}, \vec{b} = \begin{pmatrix} 5 \\ 4 \\ 0 \end{pmatrix} \quad \dots \text{ weitere Beispiele } \dots$$

- Berechnen Sie die Beträge der Vektoren  $\vec{a}$  und  $\vec{b}$  sowie ihr Produkt  $|\vec{a}| \cdot |\vec{b}|$ .
- Berechnen Sie das Skalarprodukt der Vektoren  $\vec{a}$  und  $\vec{b}$ .
- Vergleichen Sie das Skalarprodukt  $\langle \vec{a}, \vec{b} \rangle$  mit dem Produkt der Beträge  $|\vec{a}| \cdot |\vec{b}|$ .
- Stellen Sie die Vektoren  $\vec{a}$  und  $\vec{b}$  mithilfe von POV-Ray dar (als Pfeile, die im Ursprung beginnen); verwenden Sie die Anweisung `ortsvektor (<x,y,z>, Textur einfügen)`.<sup>122</sup>
- Betrachten Sie die grafische Darstellung aus verschiedenen Richtungen. Schätzen Sie den Winkel zwischen den beiden Vektoren.
- Können Sie einen Zusammenhang zwischen dem Winkel der beiden Vektoren und dem Skalarprodukt erkennen? Ist das Produkt der Beträge dabei von Bedeutung?

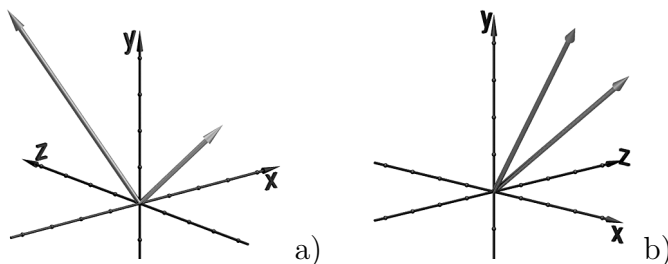


Abbildung 4.47: Lösungen zu Aufgabe A:

$$\text{a) } \langle \vec{a}, \vec{b} \rangle = 3, \quad |\vec{a}| \cdot |\vec{b}| \approx 18,5$$

$$\text{b) } \langle \vec{a}, \vec{b} \rangle = 54, \quad |\vec{a}| \cdot |\vec{b}| \approx 57,6$$

<sup>122</sup>Dieser Hinweis geht davon aus, dass die Schüler bereits Vektoren mithilfe von POV-Ray dargestellt haben (vgl. 4.2) und ist natürlich zu verändern, wenn im Unterricht eine andere Software genutzt wird.

Im Ergebnis der Lösung derartiger Aufgaben gelangen die Schüler zu der Erkenntnis, dass der Winkel zwischen zwei Vektoren klein ist, wenn der Absolutbetrag des Skalarproduktes annähernd so groß ist wie das Produkt der Beträge der Vektoren,<sup>123</sup> und der Winkel vergleichsweise groß wird, wenn das Skalarprodukt im Verhältnis zum Produkt der Beträge einen geringen Betrag hat. Damit liegt es nahe, dass der Winkel zweier Vektoren mit dem Quotienten  $\frac{\langle \vec{a}, \vec{b} \rangle}{|\vec{a}| \cdot |\vec{b}|}$  in Beziehung steht, über den anhand der bisherigen Überlegungen leicht folgende Tatsachen herausgearbeitet werden können:

$$-1 \leq \frac{\langle \vec{a}, \vec{b} \rangle}{|\vec{a}| \cdot |\vec{b}|} \leq 1; \quad \frac{\langle \vec{a}, \vec{b} \rangle}{|\vec{a}| \cdot |\vec{b}|} = 1, \text{ falls } \angle(\vec{a}, \vec{b}) = 0; \quad \frac{\langle \vec{a}, \vec{b} \rangle}{|\vec{a}| \cdot |\vec{b}|} = 0, \text{ falls } \angle(\vec{a}, \vec{b}) = 90^\circ.$$

Natürlich folgt daraus noch nicht der Zusammenhang

$$\cos \angle(\vec{a}, \vec{b}) = \frac{\langle \vec{a}, \vec{b} \rangle}{|\vec{a}| \cdot |\vec{b}|} \quad \text{bzw.} \quad \langle \vec{a}, \vec{b} \rangle = |\vec{a}| \cdot |\vec{b}| \cdot \cos \angle(\vec{a}, \vec{b}), \quad (4.2)$$

aber die Schüler können zumindest nach einer Funktion suchen, die den Zusammenhang zwischen dem Winkel zweier Vektoren und dem Quotienten  $\frac{\langle \vec{a}, \vec{b} \rangle}{|\vec{a}| \cdot |\vec{b}|}$  beschreibt und den herausgearbeiteten Bedingungen gerecht wird, wobei evtl. der Hinweis zu geben ist, die Eigenschaften der trigonometrischen Funktionen zu betrachten.<sup>124</sup> Aufbauend auf derartigen Überlegungen lässt sich die Gleichung (4.2) herausarbeiten. Ein Beweis (der mithilfe des Kosinussatzes geführt werden kann, siehe z. B. [4], S. 129f., [208], S. 106f., [224], S. 169 und [247], S. 185f.) sollte zumindest in Leistungskursen erfolgen; in Grundkursen könnte der Beweis eine Aufgabe für leistungsstärkere Schüler sein.

### 4.6.3 Normalenvektoren von Ebenen

Um einen Normalenvektor  $\vec{n}$  einer durch eine Parameterdarstellung gegebenen Ebene zu ermitteln, muss ein Vektor bestimmt werden, der zu beiden Richtungsvektoren  $\vec{a}$  und  $\vec{b}$  der Ebene orthogonal ist. Steht das Vektorprodukt (welches nach den meisten Rahmenplänen nicht behandelt wird) nicht zu Verfügung, so kann dies wegen (4.1) durch Lösen des Gleichungssystems

$$\begin{aligned} x_a x_n + y_a y_n + z_a z_n &= 0 \\ x_b x_n + y_b y_n + z_b z_n &= 0 \end{aligned}$$

(mit den Variablen  $x_n, y_n, z_n$  und den durch die Richtungsvektoren  $\vec{a}$  und  $\vec{b}$  gegebenen Koeffizienten  $x_a, y_a, z_a$  sowie  $x_b, y_b, z_b$ ) erfolgen. Einfacher ist die Bestimmung von Normalenvektoren für Ebenen, die durch Koordinatengleichungen gegeben sind. Die Schüler müssen hierfür nur erkennen, dass der Koeffizientenvektor der die Ebene beschreibenden Gleichung senkrecht auf der Ebene steht. Dazu kann eine Aufgabe folgender Art gestellt werden:

<sup>123</sup>Dass der Absolutbetrag des Skalarproduktes in keinem Falle größer als das Produkt der Beträge sein kann, lässt sich leicht überlegen und bereits vor der Einführung des Begriffs „Skalarprodukt“ anhand des Terms  $x_a x_b + y_a y_b + z_a z_b$  erarbeiten.

<sup>124</sup>In einem Grundkurs (vgl. 5.1) gelangten zumindest einige Schüler nach diesem Hinweis zu der Vermutung, dass  $\frac{\langle \vec{a}, \vec{b} \rangle}{|\vec{a}| \cdot |\vec{b}|}$  gleich dem Kosinus des Winkels zwischen den Vektoren  $\vec{a}$  und  $\vec{b}$  sein könnte.

B. Gegeben ist eine Ebene  $E$  durch eine Gleichung der Form  $Ax + By + Cz + D = 0$ .

a)  $E : 2x + 3y + 4z - 4 = 0$       b)  $E : 2x - 1,5y - 3z + 4 = 0$

- Ermitteln Sie einen Punkt  $P$ , der in der Ebene  $E$  liegt.
- Stellen Sie die Ebene  $E$  durch die Anweisung `ebene (A,B,C,D, Textur)` und den von Ihnen ermittelten Punkt mittels `punkt (<x,y,z>, Textur)` dar.
- Stellen Sie nun außerdem den Vektor  $\begin{pmatrix} A \\ B \\ C \end{pmatrix}$ , der sich aus den Koeffizienten  $A$ ,  $B$  und  $C$  der Ebenengleichung ergibt, als Pfeil dar, der im Punkt  $P$  beginnt:  
`vektoranpunkt (<xP,yP,zP`
- Betrachten Sie die Darstellung aus verschiedenen Richtungen. Welche Vermutung haben Sie hinsichtlich der gegenseitigen Lage der Ebene  $E$  und des Vektors  $\begin{pmatrix} A \\ B \\ C \end{pmatrix}$ ?

Abbildung 4.48 zeigt zwei Grafiken, die bei der Lösung der Aufgabe angefertigt wurden. Bei Betrachtung der Darstellungen aus unterschiedlichen Perspektiven wird deutlich, dass die Koeffizientenvektoren orthogonal zu den entsprechenden Ebenen sind.

Es lässt sich nun recht schnell herausarbeiten, dass eine Ebene durch einen ihr angehörenden Punkt und einen Normalenvektor eindeutig bestimmt ist, wovon ausgehend Normalengleichungen von Ebenen behandelt werden können. Falls die Schüler mit dem CAS MuPAD arbeiten, können sie jetzt Ebenen als **Plane**-Objekte darstellen, bei denen ein Punkt  $P(x; y; z)$  und ein Normalenvektor  $\vec{n}$  der Ebene vorzugeben sind:<sup>125</sup>

`plot::Plane([x,y,z], [nx,ny,nz]).`

<sup>125</sup>Bei der in Abschnitt 4.2.3 beschriebenen Darstellung von Ebenen mithilfe von MuPAD konnte die **Plane**-Anweisung noch nicht genutzt werden, da diese das Verständnis der Beschreibung von Ebenen durch Punkte und Normalenvektoren voraussetzt. Nach der Behandlung von Normalenvektoren ist es sinnvoll, die in 4.2.3 beschriebenen Möglichkeiten zur Darstellung von Ebenen (**Surface**, **Implicit3D** und **Parallelogram3d**, vgl. S. 242) mit der **Plane**-Anweisung in Beziehung zu setzen. Die Schüler können die unterschiedlichen Befehle zur Darstellung von Ebenen nutzen, um bei Aufgaben zur Umwandlung verschiedener Arten von Ebenengleichungen die Richtigkeit ihrer Ergebnisse zu überprüfen.

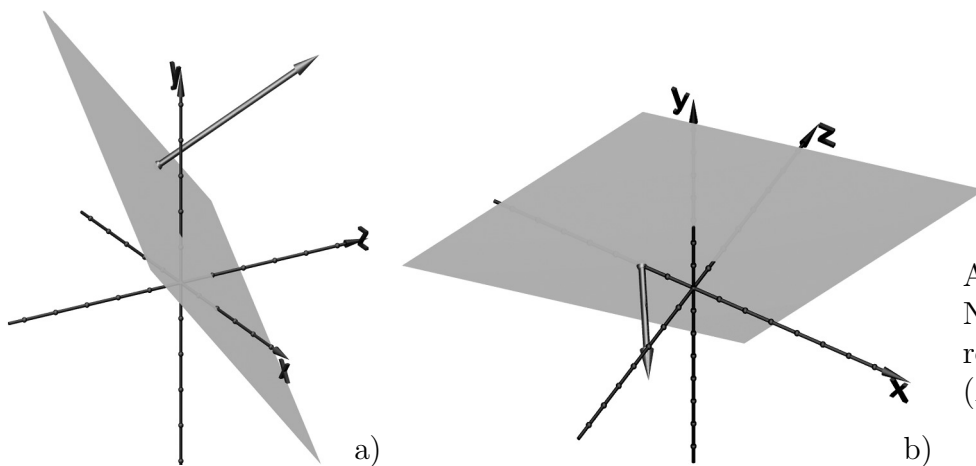


Abbildung 4.48:  
Normalenvektoren von Ebenen  
(Aufgabe B)

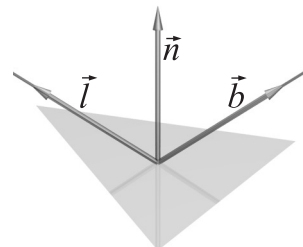
### 4.6.4 Aufgaben zum räumlichen Reflexionsgesetz

Nach der Behandlung des Skalarproduktes sowie von Normalenvektoren lässt sich die eingangs dieses Abschnitts (siehe S. 275) aufgeworfene Frage nach einer mathematischen Beschreibung der Reflexion im Raum wieder aufgreifen. Die Schüler können jetzt das Reflexionsgesetz z. B. folgendermaßen formulieren:

*Der Winkel zwischen einem Richtungsvektor des einfallenden Lichtstrahls und einem Normalenvektor sowie der Winkel zwischen einem Richtungsvektor des reflektierten Lichtstrahls und dem Normalenvektor sind maßgleich; die drei Vektoren sind komplanar.*<sup>126</sup>

Eine vergleichsweise einfache Formulierung des Reflexionsgesetzes und die Möglichkeit, Berechnungen damit effektiv durchzuführen, ergeben sich, wenn Normaleneinheitsvektoren  $\vec{n}$  sowie normierte Verbindungsvektoren  $\vec{l}$  und  $\vec{b}$  zwischen dem Auftreffpunkt des Lichtstrahls auf die Oberfläche und der Lichtquelle bzw. der Kamera verwendet werden. Das Reflexionsgesetz lässt sich dann in der bereits auf S. 128 angegebenen Form, also durch die beiden folgenden Bedingungen, ausdrücken:

- $\vec{n}$ ,  $\vec{l}$  und  $\vec{b}$  sind komplanar,
- $\langle \vec{l}, \vec{n} \rangle = \langle \vec{b}, \vec{n} \rangle$ .



Mithilfe von POV-Ray können die Schüler einige Anwendungsaufgaben zum Reflexionsgesetz bearbeiten, die gleichzeitig der Festigung der Kenntnisse zum Skalarprodukt und zu Normalenvektoren dienen. Wenn bereits (wie in Abschnitt 4.6.1 vorgeschlagen) verdeutlicht wurde, dass „reale“ Objekte in der 3D-Computergrafik meist durch Dreiecksnetze dargestellt werden, so ergibt sich daraus eine Rechtfertigung, Betrachtungen zur Reflexion anhand von Dreiecken zu führen.<sup>127</sup> Hierfür eignen sich z. B. die beiden folgenden Aufgaben C und D.<sup>128</sup>

<sup>126</sup>Die Komplanaritätsbedingung vergessen die Schüler erfahrungsgemäß leicht, was daran liegen könnte, dass diese Bedingung bei der Behandlung des Reflexionsgesetzes im Physikunterricht (siehe S. 275) oft nicht erwähnt, sondern als selbstverständlich vorausgesetzt wird. Bei der Formulierung des Reflexionsgesetzes für die Lichtausbreitung im Raum ist darauf jedoch Wert zu legen.

<sup>127</sup>Natürlich könnten die in den Aufgaben vorzunehmenden Untersuchungen auch anhand von Ebenen erfolgen. Die Ergebnisse sind jedoch an relativ kleinen Dreiecken aufgrund des Kontrasts zum Hintergrund besser zu erkennen; außerdem ist für in Abschnitt 4.6.6 beschriebene komplexere Aufgaben die Konstruktion von Objekten aus Dreiecken notwendig.

<sup>128</sup>Für die Bearbeitung dieser Aufgaben werden einige der in den Include-Dateien `anageoL.inc` bzw. `anageoR.inc` definierten Objekte und Texturen benötigt (siehe Abschnitt 4.2 sowie die Anleitung auf S. 335f.). Allerdings stören bei Experimenten zur Reflexion und Beleuchtung von Objekten die dort vordefinierten Lichtquellen; auch die Kameraposition sollen die Schüler bei den folgenden Aufgaben selbst durch Koordinaten festlegen. Da zudem die in den `anageo`-Makros definierten Objekte mit Standardobjekten von POV-Ray zu kombinieren sind, kann nur die Variante mit linkshändigem KS genutzt werden. Eine geeignete Version (ohne Lichtquellen und Kamera) ist die Datei `anageoL-ohneKam.inc` (Ordner `Vorlagen+Anleitungen\Vorlagen`, siehe S. 365f.). Vor der Bearbeitung der Aufgaben C und D sollten die Schüler eine Szene mit einer Kamera und einer Lichtquelle erstellen, deren Positionen

C. Gegeben ist die Ebene  $E$  durch die Punkte  $A(1; 1; -1)$ ,  $B(-1; 0; 2)$  und  $C(0; 0; -1)$ .

- Bestimmen Sie einen Normaleneinheitsvektor der Ebene  $E$ .
- Stellen Sie in POV-Ray das Dreieck mit den Eckpunkten  $A$ ,  $B$  und  $C$  dar:  
`object { triangle { <1,1,-1>, <-1,0,2>, <0,0,-1> } Textur einfügen }.`  
 Geben Sie dem Dreieck eine spiegelnde Oberfläche, indem Sie z. B. die Textur `silbergrau` verwenden.
- Bestimmen Sie die Koordinaten des Schwerpunktes  $S$  des Dreiecks und stellen Sie den Normaleneinheitsvektor der Ebene als Pfeil dar, der in  $S$  beginnt. Verwenden Sie dazu den Befehl `verbindungsvektor`, indem Sie zunächst berechnen, in welchem Punkt sich die Pfeilspitze befinden muss.
- Bestimmen Sie den Verbindungsvektor des Schwerpunktes  $S$  mit der Kameraposition und normieren Sie diesen Vektor (teilen Sie ihn durch seinen Betrag). Stellen Sie jetzt diesen „Kameraeinheitsvektor“ ebenfalls als Pfeil, der in  $S$  beginnt, dar.
- Fügen Sie mittels `light_source { <x,y,z> color White }` eine Lichtquelle ein und berechnen Sie den Verbindungsvektor von  $S$  zur Position dieser Lichtquelle. Normieren Sie diesen Vektor ebenfalls und stellen Sie auch den dabei erhaltenen „Lichtquelleneinheitsvektor“ als Pfeil dar, der in  $S$  beginnt.  
 Geben Sie den drei Vektoren unterschiedliche Farben, damit Sie diese gut unterscheiden können. Warum sehen Sie von dem „Kameraeinheitsvektor“ fast nichts (nur einen kleinen Kreis, der die Pfeilspitze verdeutlicht)?
- Verändern Sie die Kameraposition, um die Lage der drei Vektoren besser erkennen zu können. Der Kameravektor bezieht sich dann aber nicht mehr auf die verwendete (neue) Kamera – er ist dadurch besser sichtbar.

D. Versuchen Sie, die Lichtquelle in der Szene aus Aufgabe C so zu positionieren, dass das Dreieck so hell wie möglich beleuchtet wird.

- Damit Sie die Position der Lichtquelle im Bild erkennen können, fügen Sie der Szene eine kleine Kugel hinzu, deren Mittelpunkt dieselben Koordinaten wie die

sie dann variieren. Haben die Schüler vorher in POV-Ray Kameras und Lichtquellen noch nicht selbst beschrieben, so können sie aus Gründen der Zeitersparnis auch eine entsprechend vorbereitete Datei wie z. B. `Reflexionsexperimente-Vorlage.pov` (in dem Ordner 4-6) verwenden, in der sie nur die Koordinaten der Kamera und einer Lichtquelle verändern müssen.

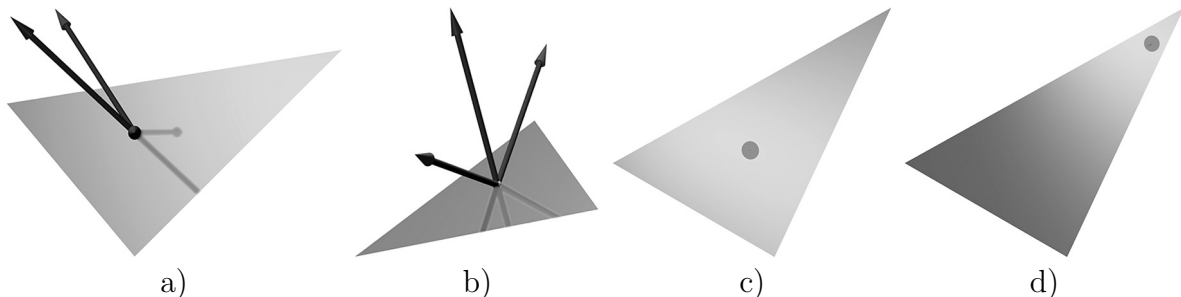


Abbildung 4.49: Lösungsgrafiken zu den Aufgaben C (a, b) und D (c, d)



Lichtquelle hat, z. B. mittels

```
sphere { <Koordinaten der Lichtquelle> 0.1 texture{rot_glanz} no_shadow }.
```

Die Anweisung **no\_shadow** ist wichtig, da die Kugel ansonsten das gesamte Licht der Lichtquelle in ihrem Inneren „gefangen halten“ würde; mit der **no\_shadow**-Anweisung kann das Licht die Kugel ungehindert durchdringen.

- Verändern Sie jetzt die Koordinaten der Lichtquelle und der kleinen Kugel so, dass Sie das Spiegelbild der Kugel auf dem Dreieck sehen. Betrachten Sie die Helligkeit der Dreiecksfläche und die Position des Spiegelbildes.

Anhand der Aufgaben C und D können die Schüler Normaleneinheitsvektoren von Ebenen berechnen und veranschaulichen sowie Winkel zwischen Vektoren abschätzen. Diese Aufgaben kombinieren somit gebräuchliche Standardaufgaben mit visuell-experimentellen Vorgehensweisen.<sup>129</sup> Zudem werden Aspekte der Beleuchtung in den Mittelpunkt des Interesses gerückt, deren Vertiefung Gegenstand der folgenden Überlegungen ist.

#### 4.6.5 Lokale Beleuchtungsmodelle

Sowohl anhand der Sichtbarkeit von Objekten in der realen Natur als auch bei den im vorangegangenen Abschnitt behandelten Aufgaben C und D wird deutlich, dass Punkte von Objektoberflächen nicht nur dann sichtbar sind, wenn für sie (in Bezug auf die Kamera und eine in der Szene vorhandene Lichtquelle) das Reflexionsgesetz erfüllt ist. In diesem Falle müssten alle Punkte, für welche ein in ihnen errichteter Normaleneinheitsvektor  $\vec{n}$  der betrachteten Fläche sowie die normierten Verbindungsvektoren  $\vec{l}$  und  $\vec{b}$  zur Lichtquelle und zur Kamera nicht komplanar sind bzw. für die nicht  $\langle \vec{l}, \vec{n} \rangle = \langle \vec{b}, \vec{n} \rangle$  gilt, aus Sicht der Kamera völlig unbeleuchtet, also schwarz sein. Da dies offensichtlich nicht der Fall ist, werden Objekte in der Natur und in 3D-Grafikprogrammen nicht nur durch exakte Reflexion beleuchtet. Ausgehend von dieser Überlegung können die wichtigsten lokalen Beleuchtungsmodelle (siehe Abschnitt 2.5.2) behandelt werden, wobei folgende Aspekte ineinander greifen:

- Die Schüler erarbeiten – nach einer Diskussion verschiedener Wege der Lichtausbreitung in der realen Natur, entsprechenden Computerexperimenten sowie der Formulierung vereinfachter Realmodelle<sup>130</sup> – mathematische Modelle unter Anwendung des Skalarproduktes sowie von Normalenvektoren.
- Sie erlangen ein Verständnis der bei der praktischen Anwendung von 3D-Grafiksoftware für die Oberflächengestaltung wichtigen Parameter **ambient**, **diffuse**,

<sup>129</sup>POV-Ray-Dateien mit möglichen Lösungen der Aufgaben sowie die in Aufgabe C geforderten Berechnungen (als PDF-Datei) enthält der Ordner 4-6.

<sup>130</sup>Die Zusammenhänge zwischen realen Situationen, Realmodellen und mathematischen Modellen bei Modellierungsprozessen im Mathematikunterricht werden u. a. in [110], S. 1–5 beschrieben (siehe auch Abschnitt 1.4.6 der vorliegenden Arbeit). Natürlich stellen auch die Annahmen der Lichtausbreitung, die u. a. den Modellen der diffusen Streuung und der spekularen Glanzlichter (nach PHONG) zugrunde liegen, bereits Vereinfachungen der äußerst komplexen Reflexions- und Streuvorgänge in der realen Natur dar. Derartige Vereinfachungen (Idealisierungen) sind notwendig, um komplexe reale Situationen überhaupt mathematisch beschreiben zu können.

`phong`, `phong_size` sowie `reflection` und können sie dadurch in Abhängigkeit vom gewünschten Erscheinungsbild einer Oberfläche geeignet justieren.

Die Überlegungen zu Beleuchtungskomponenten können durch praktische Experimente mithilfe von POV-Ray beginnen, indem die Schüler ein bis zwei Lichtquellen sowie mindestens zwei geometrische Objekte erstellen,<sup>131</sup> diesen mittels

```
texture { pigment { color rgb <r,g,b> }132
      finish { ambient  $k_a$  diffuse  $k_d$  phong  $k_p$  reflection  $k_r$  }133
```

Texturen zuweisen und die Werte der angegebenen Renderparameter (zwischen 0 und 1) variieren. Insbesondere sollten sie versuchen, die Bedeutung dieser Parameter zu erkennen, indem sie jeweils nur einen der Werte für `ambient`, `diffuse`, `phong` und `reflection` untersuchen und die anderen Werte dabei auf Null setzen. Während sich die Wirkung der ambienten Beleuchtungskomponente (welche Punkten völlig unabhängig von ihrer Lage Helligkeitswerte zuweist, vgl. S. 131) sehr schnell zeigt (siehe den linken Kegel in Abb. 4.50) und die direkte Reflexion bereits zuvor besprochen wurde, bieten vor allem die diffuse Reflexion (Lichtstreuung) und die mittels `phong` generierten Glanzpunkte interessante Ansatzpunkte für die Behandlung im Unterricht.

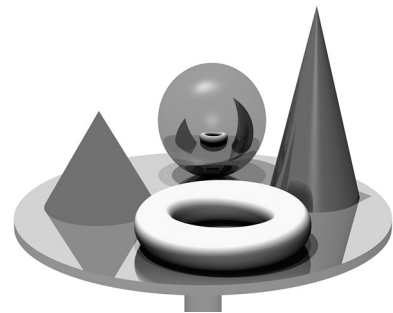


Abbildung 4.50: Objekte mit unterschiedlichen Werten der Renderparameter `ambient`, `diffuse`, `phong` und `reflection`

Hinsichtlich der *diffusen Beleuchtungskomponente* wird durch Experimente deutlich, dass die Helligkeitsverteilung auf einer Oberfläche von den Winkeln abhängt, unter denen von Lichtquellen ausgehende Strahlen auf die Oberfläche treffen. Hingegen ist

<sup>131</sup>Mindestens zwei geometrische Objekte sind notwendig, damit Spiegelungen in der Szene auftreten. Die Anordnung der Objekte und der Kamera muss so erfolgen, dass Spiegelungen eines Objekts in einem anderen sichtbar sind. Dabei können die Schüler ihre bei der Lösung von Aufgabe D (S. 282) gesammelten Erfahrungen nutzen. Das Objekt, dessen Oberflächenerscheinung hauptsächlich untersucht wird, sollte gekrümmt sein (z. B. eine Kugel), damit sich auf seiner Oberfläche die Normalenrichtungen ändern, was insbesondere für Glanzpunkte von Bedeutung ist. Eine Möglichkeit für die hier beschriebenen Betrachtungen bestünde in der Nutzung eines von den Schülern bereits zuvor modellierten Schneemanns (siehe 4.1), der durch eine Grundebene (z. B. mit Schachbrettmuster) ergänzt werden sollte. Andere geeignete Szenen sind z. B. eine Kugel, die auf einer Schachbrettebene liegt, sowie einfache Szenen wie in Abb. 4.50 oder in Abb. B.6 auf S. 350. Beispielszenen, anhand derer die Beleuchtungskomponenten gut untersucht werden können, befinden sich in dem Ordner 4-6. Allerdings erscheint es sinnvoller, die Schüler selbst entsprechende Szenen anfertigen zu lassen, was auch als Hausaufgabe erfolgen kann. Auch die beschriebenen ersten Experimente zur Wirkung der Renderparameter können von den Schülern selbstständig durchgeführt werden.

<sup>132</sup>Die Beschreibung von Farben durch RGB-Vektoren ist nur dann sinnvoll, wenn vorher im Unterricht darauf eingegangen wurde (siehe 3.3.1) oder dies zumindest kurz an dieser Stelle erfolgt. Ansonsten können auch Farbnamen verwendet werden, die in der mit POV-Ray gelieferten Datei `colors.inc` definiert sind und in der POV-Ray-Hilfe erläutert werden (Hilfeindex → `colors.inc`).

<sup>133</sup>Auf den Parameter `phong_size` (siehe S. 132f.), der die Größe der Glanzpunkte bestimmt, wird zunächst verzichtet, da seine Variation erst sinnvoll ist, wenn die Glanzpunkte thematisiert wurden.

sie von den Winkeln zwischen Oberflächennormalen und Sehstrahlen unabhängig. Da hauptsächlich diffus beleuchtete Flächen recht matt und rau wirken, liegt es nahe, dass es sich um unregelmäßig geformte Oberflächen handelt, an deren Mikrofacetten Lichtstrahlen in unterschiedliche Richtungen reflektiert werden. Davon ausgehend und unter Beachtung der Fläche, auf die das von der Lichtquelle ausgehende Licht verteilt wird, lässt sich – wie in Abschnitt 2.5.2, S. 131f. beschrieben – das LAMBERTSche Kosinusetz erarbeiten. Die Schüler sollten dabei den Zusammenhang zwischen der Beleuchtungsintensität und dem Skalarprodukt des Normaleneinheitsvektors und des Verbindungsvektors mit der Lichtquelle möglichst selbst entwickeln.<sup>134</sup>

Den Ausgangspunkt für die Betrachtung der PHONGschen *Beleuchtungskomponente* bilden Glanzpunkte, die z. B. auf Metallgegenständen oder Billardkugeln sichtbar sind und sich auf mit POV-Ray generierten Oberflächen zeigen, wenn ein recht großer Wert für den Renderparameter `phong` eingestellt wird (siehe Abb. B.18 d) und e) auf S. 353). Als von Schülern gegebene Erklärung für das Auftreten der Glanzpunkte ist zu erwarten, dass sich die Lichtquellen in den Objektoberflächen spiegeln. Allerdings widersprechen ausgedehnte Glanzpunkte dem in der Computergrafik gebräuchlichen Modell punktförmiger Lichtquellen, so dass dieses Phänomen nicht durch das Reflexionsgesetz zu erklären ist, sondern eine „unscharfe Reflexion“ erfolgen muss.

Die Schüler können in ihren Dateien, anhand derer sie mit verschiedenen Renderparametern experimentiert haben, nun den zusätzlichen Parameter `phong_size` einfügen und variieren, wobei sich auch recht ausgedehnte Leuchtflecken erzeugen lassen. Dabei machen sie die scheinbar paradoxe Feststellung, dass kleine Werte für `phong_size` größere, große Werte hingegen kleine, scharf abgegrenzte Glanzpunkte hervorrufen.

Um das Phänomen der „unscharfen Reflexion“, die Glanzpunkte verursacht, zu beschreiben, sollte nochmals das Reflexionsgesetz betrachtet werden. In anderer Formulierung besagt es, dass direkte Reflexion genau dann auftritt, wenn die Richtung des Normaleneinheitsvektors in einem Oberflächenpunkt mit der Richtung der Winkelhalbierenden zwischen den normierten Verbindungsvektoren  $\vec{l}$  und  $\vec{b}$  des Punktes mit der Lichtquelle bzw. der Kamera identisch ist. Dies lässt sich auch folgendermaßen ausdrücken:

$$\vec{n} = \frac{\vec{b} + \vec{l}}{|\vec{b} + \vec{l}|}, \quad (4.3)$$

siehe auch die Fassung (2.15) des Reflexionsgesetzes auf S. 129. Wird der normierte Richtungsvektor  $\frac{\vec{b} + \vec{l}}{|\vec{b} + \vec{l}|}$  der Winkelhalbierenden mit  $\vec{h}$  (Highlight-Vektor) bezeichnet, so trifft die Gleichung (4.3), also  $\vec{n} = \vec{h}$ , genau dann zu, wenn das Skalarprodukt des Highlight- mit dem Normaleneinheitsvektor Eins ist. Diese Bedingung lässt sich dadurch „entscharfen“ (und somit unscharfe Reflexion, die zu Glanzflecken führt, simulieren), dass auch für Punkte mit Skalarprodukten  $\langle \vec{n}, \vec{h} \rangle < 1$  Beleuchtungsanteile hinzugefügt

<sup>134</sup>Alternativ oder in Ergänzung zu der auf S. 131f. beschriebenen Herleitung lässt sich die Proportionalität zwischen dem Kosinus des Lichteinfallswinkels und der Beleuchtungsintensität (und somit das LAMBERTSche Kosinusetz) auch experimentell mithilfe einer drehbar gelagerten Solarzelle erarbeiten; einen entsprechenden Vorschlag beschrieb OLDENBURG in [194].

werden. Diese müssen allerdings mit der Stärke der Verletzung der Bedingung  $\langle \vec{n}, \vec{h} \rangle = 1$  für die ideale Reflexion, d. h. mit abnehmendem Skalarprodukt, recht schnell an Intensität verlieren. Ein einstellbarer Intensitätsverlust lässt sich durch den Faktor

$$\langle \vec{n}, \vec{h} \rangle^m$$

für die Intensität der PHONGschen Beleuchtungskomponente erreichen, der das „Kernstück“ der Simulation von Glanzflecken bildet (siehe S. 132f.). Entscheidend für Größe und Verlauf der Glanzflecken ist die Abhängigkeit der Lichtintensität vom Winkel zwischen  $\vec{n}$  und  $\vec{h}$ . Die Schüler können die entsprechende Funktion für verschiedene Exponenten  $m$  untersuchen (siehe Abb. 2.71 auf S. 133) sowie Grafiken unter Verwendung der betrachteten Exponenten (in POV-Ray `phong_size`) anfertigen. Das Paradoxon kleinerer Glanzflecken bei größeren Werten für `phong_size` klärt sich dabei auf.

Anhand der skizzierten Betrachtungen zu Beleuchtungskomponenten<sup>135</sup> lassen sich interessante Überlegungen zum Verhältnis von Realsituationen und ihrer Idealisierung zu Realmodellen, daraus erarbeiteten mathematischen Modellen sowie deren Anwendung für Computersimulationen führen (siehe auch S. 49f.). Im Ergebnis verfügen die Schüler über Voraussetzungen, um Oberflächenerscheinungen zu simulieren, wobei es sich um einen zentralen (und für viele Schüler aufgrund der ästhetischen Komponente reizvollen) Aspekt der 3D-Computergrafik handelt.<sup>136</sup> Hervorgehoben werden sollte abschließend die Bedeutung der Oberflächennormalen für Beleuchtungssimulationen, auf die bei den im Folgenden geschilderten Überlegungen noch zurückgekommen wird.

#### 4.6.6 Kantenglättung durch Normaleninterpolation

Angesichts der anhand des Reflexionsgesetzes und der Beleuchtungsmodelle deutlich gewordenen Bedeutung der Normalenvektoren für das Erscheinungsbild von Oberflächen verwundert es nicht mehr, dass durch Dreiecksnetze repräsentierte Objekte erkennbare Kanten an den Objektoberflächen aufweisen (siehe Abb. B.21 auf S. 354). Viele Schüler kennen jedoch bereits Figuren aus Computerspielen, die ebenfalls als Dreiecksnetze aufgebaut und auf deren Oberflächen dennoch keine Kanten sichtbar sind.<sup>137</sup> Als Vorgriff auf die folgenden Betrachtungen können den Schülern die POV-Ray-Beschreibungen des Dreiecksnetzes eines „realen“ Objekts, z. B. des auf S. 354 abgebildeten Fisches,<sup>138</sup> in einer Version mit sichtbaren Kanten und einer mittels Normaleninterpolation „geglättete-

<sup>135</sup>Bei Zeitknappheit wäre der Verzicht auf die Behandlung der (hinsichtlich ihrer mathematischen Beschreibung anspruchsvolleren und mehr Zeit erfordernden) PHONGschen Komponente und eine Beschränkung auf die diffuse Beleuchtung ins Auge zu fassen.

<sup>136</sup>Die Nutzung des erlangten Wissens für die Perfektionierung der Grafiken der Schüler wird sicherlich nicht mehr im Unterricht erfolgen. Es ist jedoch zu erwarten, dass zumindest einige Schüler in der Freizeit weiter an ihren Grafiken arbeiten und dabei der Gestaltung von Oberflächen durch gezielten Einsatz der Renderparameter Aufmerksamkeit widmen.

<sup>137</sup>Dass in der Computergrafik Dreiecksnetze von hoher Bedeutung sind, könnte Schülern, die sich für Computerspiele begeistern, u. a. dadurch bekannt sein, dass in der Werbung für spieletaugliche Grafikkarten die Anzahl der je Sekunde verarbeiteten Dreiecke ein wichtiges Argument darstellt.

<sup>138</sup>Die Objektbeschreibungen `FISH.INC` und `FISHG1att.INC` des Fisches (ohne bzw. mit Eckpunktnormalen) befinden sich in dem Ordner 2-5 (siehe S. 365f.).

ten“ Version vorgelegt werden. Diese unterscheiden sich dadurch, dass die ungeglättete Version aus (sehr vielen) Zeilen mit Beschreibungen von Dreiecken in der Form

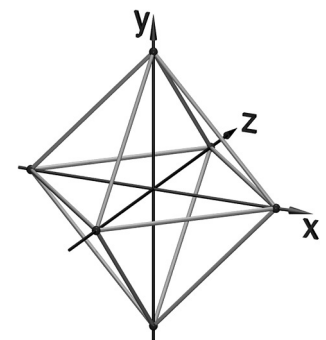
```
triangle { <-0.2214, 4.3819, 2.1779>,
           <-0.2665, 4.2580, 1.9298>,
           <-0.3268, 4.5006, 1.8141> }
```

besteht, in der geglätteten Version Dreiecke hingegen durch jeweils 6 Koordinatentripel beschrieben werden:

```
smooth_triangle { <-0.2214, 4.3819, 2.1779>, <-0.916, -0.175, 0.362>,
                  <-0.2665, 4.2580, 1.9298>, <-0.900, -0.338, 0.275>,
                  <-0.3268, 4.5006, 1.8141>, <-0.978, -0.042, 0.206> } .
```

Nach einer Besprechung des Prinzips der Kantenglättung durch Normaleninterpolation (siehe Abschnitt 2.5.3), welche für jeden Eckpunkt die Berechnung eines aus den Normalen der angrenzenden Dreiecke gemittelten Einheitsvektors erfordert, können Schüler dieses „Kantenglättungsverfahren“ anhand eines einfachen, recht extremen und deshalb umso verblüffenderen Beispiels selbst vollziehen. Dazu lässt sich die Aufgabe stellen, ein Oktaeder durch seine dreieckigen Seitenflächen darzustellen, die Normaleneinheitsvektoren aller Ebenen, in denen die Seitenflächen liegen, zu bestimmen und schließlich für alle Eckpunkte aus den Normaleneinheitsvektoren der anliegenden Seitenflächen gemittelte Vektoren zu berechnen, zu normieren und den Eckpunkten zuzuweisen.

E. Die nebenstehende Abbildung zeigt ein Oktaeder (Achtflächner) mit den Eckpunkten  $P_1(1;0;0)$ ,  $P_2(0;1;0)$ ,  $P_3(-1;0;0)$ ,  $P_4(0;-1;0)$ ,  $P_5(0;0;-1)$  und  $P_6(0;0;1)$ .



- Geben Sie die Koordinaten der Eckpunkte des Oktaeders in POV-Ray ein: `#declare P1 = <1,0,0> ; ...`
- Überlegen Sie, welche Dreiecke Sie durch ihre Eckpunkte beschreiben müssen, um alle Seitenflächen des Oktaeders darzustellen. Stellen Sie das Oktaeder in POV-Ray als Vereinigung von Dreiecken folgendermaßen dar:

```
object { union { triangle {P1,P5,P2} ... // Insgesamt 8 Dreiecke für die
                                }           // Seitenflächen des Oktaeders
          texture { }           // Geben Sie eine Textur an
        }
```

Rendern Sie nun das Oktaeder.<sup>139</sup>

- Ermitteln Sie Gleichungen der 8 Ebenen, in denen die Seitenflächen des Oktaeders liegen.
- Bestimmen Sie für jede dieser Ebenen einen Normaleneinheitsvektor.
- Berechnen Sie für jeden der Eckpunkte  $P_1$  bis  $P_6$  des Oktaeders den Mittelwert der Normaleneinheitsvektoren aller Seitenflächen, die an den Eckpunkt angrenzen.
- Normieren Sie diese „gemittelten“ Normalenvektoren.

<sup>139</sup>Ein mögliches Ergebnis der Bildberechnung in POV-Ray zeigt Abb. 4.51 a) auf S. 288.

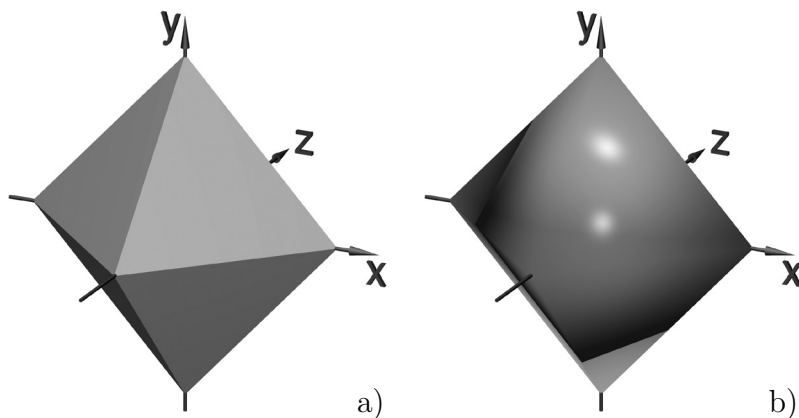


Abbildung 4.51:

- a) Darstellung eines Oktaeders durch seine Seitenflächen
- b) mittels Normaleninterpolation „geglättetes“ Oktaeder

- g) Ordnen Sie die normierten gemittelten Normalenvektoren  $\vec{n}_1 \dots \vec{n}_6$  den Punkten  $P_1 \dots P_6$  zu; stellen Sie das Oktaeder als Vereinigung „geglätteter Dreiecke“ dar:

```
object { union { smooth_triangle {P1,n1,P5,n5,P2,n2} // insgesamt 8
                                ...                // "geglättete
                                }                  // Dreiecke"
        texture { }
    }
```

Rendern Sie nun Ihre Darstellung mithilfe von POV-Ray.<sup>140</sup>

Die Aufgabe E soll lediglich exemplarisch verdeutlichen, wie Schüler (hauptsächlich durch Berechnungen, die Standardinhalte des Unterrichts sind) die Kantenglättung eines Polygonnetzes selbst vollziehen können. Es wurden sehr einfache Eckpunktkoordinaten gewählt, außerdem haben die normierten gemittelten Normalenvektoren dieselben Koordinaten wie die zugehörigen Eckpunkte.<sup>141</sup> Damit benötigen die rechnerischen Aspekte der Aufgabe nicht sehr viel Zeit. Soll die Aufgabe genutzt werden, um die auftretenden Berechnungen an schwierigeren Beispielen durchzuführen, kann ein Oktaeder mit anderen Eckpunkten vorgegeben werden; auch ein Tetraeder lässt sich heranziehen,<sup>142</sup> wobei die Anzahl gleichartiger Berechnungsaufgaben geringer ist.

In einem Unterrichtsversuch löste das als Ergebnis von Aufgabenteil g) entstandene Bild des „bauchigen“ Oktaeders (siehe Abb. 4.51 b) bei den Schülern großes Erstaunen aus. Durch Betrachtung des Bildes aus verschiedenen Positionen überzeugten sie sich davon, dass sich die Form des Oktaeders nicht verändert hat und weiche Kanten lediglich „vorgetäuscht“ werden.<sup>143</sup> Obwohl es sich natürlich um ein extremes Beispiel handelt, konnten die Schüler den Bezug zu der Glättung des zuvor als Beispiel gezeigten Fisches herstellen. Es wurde ihnen dadurch bewusst, dass sie eine wichtige Vorgehensweise der 3D-Computergrafik anhand eines einfachen Beispiels selbst vollzogen haben.

<sup>140</sup>Lösungen der Aufgabe befinden sich in dem Ordner 4-6.

<sup>141</sup>Erfahrungsgemäß finden Schüler diesen Zusammenhang nach der Berechnung von 1–3 Beispielen.

<sup>142</sup>Prinzipiell ist auch die Verwendung von Körpern mit vier- oder mehrseitigen Seitenflächen (z. B. Quadern) möglich; deren Seitenflächen müssen dazu allerdings in Dreiecke zerlegt werden.

<sup>143</sup>Besonders gut sichtbar wird diese Tatsache bei Betrachtung eines Videos, in dem sich das Oktaeder dreht bzw. die Kamera um das Oktaeder herumfährt. Ein solches Video (Oktaeder-Glatt-Video.mpg) befindet sich in dem Ordner 4-6.

## 4.7 Einbeziehung weiterer Inhalte und Anwendungen der Computergrafik

Nicht hinsichtlich aller Inhalte der Computergrafik, deren Einbeziehung in den Unterricht der analytischen Geometrie möglich und auch durchaus sinnvoll wäre, wurden in diesem Kapitel Unterrichtsvorschläge beschrieben.<sup>144</sup> Es gäbe – wie aus Kapitel 2 hervorgeht – eine Reihe weiterer Möglichkeiten, Elemente der Computergrafik zu thematisieren; im Folgenden seien einige Beispiele genannt.

- Interessant wäre u. a. die Behandlung von *Bézier- und Splinekurven sowie -flächen* (siehe die Abschnitte 2.3.6 und 2.3.7), anhand derer auch Bezüge zu für die Informatik spezifischen Vorgehensweisen gut sichtbar werden. Innerhalb eines Stoffgebietes zur analytischen Geometrie des Raumes erscheint die Beschäftigung mit Bézier- oder Splinekurven bzw. sogar -flächen im Raum allerdings nur in Ausnahmefällen geeignet, da es sich hierbei um einen recht komplexen Gegenstand handelt, dessen zweidimensionale Behandlung realistischer sein dürfte. Für die Behandlung der ebenen Bézierkurven im Unterricht liegen, wie bereits in Abschnitt 1.4.5 ausgeführt wurde, interessante und realistische Unterrichtsvorschläge vor, siehe u. a. [53], [131], [132], [179], [180], [183] und [213]. Aufgrund der Schwerpunktsetzung der vorliegenden Arbeit auf die analytische Geometrie des dreidimensionalen Raumes wird auf diese Thematik nicht näher eingegangen. Inhaltlich durchaus sinnvoll wäre es, Bézierkurven im Rahmen eines Stoffgebietes zur ebenen analytischen Geometrie zu behandeln und im Raum zur Modellierung von Rotationskörpern und auf Kurven erzeugten verallgemeinerten Zylindern anzuwenden. Dies würde jedoch insgesamt viel Zeit erfordern, die aufgrund der Curricula wohl nur in Ausnahmefällen zur Verfügung steht. Hätten die Schüler allerdings Bézierkurven entsprechend einem der erwähnten Unterrichtsvorschläge kennengelernt, so könnte darauf bei der Konstruktion von Flächen aus Kurven zurückgegriffen werden, siehe Abschnitt 2.3.7.
- Eine interessante und praktisch nachvollziehbare Anwendung analytischer Beschreibungen von Projektionen ist die Rekonstruktion von 3D-Koordinaten aus mehreren Bildern derselben Objekte, die aus verschiedenen Perspektiven aufgenommen wurden. Dazu sind Überlegungen zu Lösungsverfahren überbestimmter Gleichungssysteme anzustellen; einen entsprechenden Unterrichtsvorschlag unterbreitete OLDENBURG in [193].
- Ebenfalls auf weiter gehende Überlegungen zu Projektionen baut die Behandlung stereografischer Projektionen auf, die u. a. genutzt werden können, um mithilfe von Stereobrillen einen räumlichen Eindruck bei der Betrachtung von Bildern dreidimensionaler Szenen zu erwecken (siehe u. a. die Unterrichtsvorschläge von LEUDERS in [156] und von OLDENBURG in [191]).

---

<sup>144</sup>Wie in der Einleitung des Kapitels erwähnt wurde, erfolgte die Auswahl der Themengebiete u. a. unter dem Blickwinkel einer gewissen Nähe zu den Inhalten der Curricula. Die auf dieser Seite genannten Themen eignen sich eher für die Behandlung in Ergänzungskursen oder Arbeitsgemeinschaften.

## 4.8 Zusammenfassende Anmerkungen zur Berücksichtigung von Aspekten der Informatik

Werden Elemente der Computergrafik in den Unterricht der analytischen Geometrie einbezogen, so ist es anhand einer Reihe von Aspekten inhaltlich sinnvoll, Vernetzungen zum Fach Informatik herzustellen. Erschwerend hierfür erweist sich jedoch, dass meist nur eine Minderheit der beteiligten Schüler Informatikkurse belegt. Aufgrund der im Stoffgebiet Analytische Geometrie verfolgten Ziele können informatikspezifische Inhalte zudem nicht im Mittelpunkt des Unterrichts stehen. Andererseits lassen sich Denk- und Arbeitsweisen der Informatik in vielen Fällen nicht von denen der Mathematik trennen; so wurde bereits in Abschnitt 1.5 ausgeführt, dass Aspekte algorithmischen Arbeitens für die Mathematik und die Informatik gleichermaßen von Bedeutung sind.<sup>145</sup>

Vor allem die folgenden Inhalte und Arbeitsweisen der Informatik können bei der Einbeziehung von Elementen der 3D-Computergrafik in das Stoffgebiet Analytische Geometrie berücksichtigt werden:

- Die *Arbeit mit Variablen* ist für die Mathematik und für die Informatik von grundlegender Bedeutung. Gegenüber der im MU üblichen Variablenauffassung kommt in der Informatik ein erweiterter Variablenbegriff zur Anwendung, worauf an verschiedenen Stellen eingegangen werden kann (siehe u. a. Abschnitt 4.1.5, S. 225f.).
- Bei der Behandlung von *Vektoren* und evtl. von *Matrizen* lässt sich auf deren Bedeutung als Datenstrukturen in der Informatik eingehen (vgl. S. 194f.).
- Bei der Arbeit mit Computergrafiken lassen sich in elementarer Weise *objektorientierte Sichtweisen* herausarbeiten. So können konkrete geometrische Objekte als Instanzen von Klassen (wie **sphere**, **box** oder **cone**) mit spezifischen Attributen (wie Positionen, Ausdehnungen und Texturen) aufgefasst werden. Darauf anwendbare Methoden sind vor allem geometrische Transformationen. Etwas weiter geht die Konstruktion von „Prototypen“ (die sich als Klassen auffassen lassen) und die Erzeugung mehrerer Instanzen davon. Als elementares Beispiel hierfür lässt sich die auf S. 258 kurz beschriebene Erstellung mehrerer Exemplare eines zuvor konstruierten Schneemannes durch einen Schüler der IGH ansehen.<sup>146</sup>

Es ist einschränkend festzuhalten, dass anhand der in der vorliegenden Arbeit unterbreiteten Vorschläge lediglich grundlegende Ideen der Objektorientierung zum Tragen kommen können, die zudem auf geometrische Objekte beschränkt bleiben. Die hauptsächlich zur Verwendung vorgeschlagenen Softwarepakete POV-Ray und MuPAD zeichnen sich zudem nicht durch eine Syntax aus, die objektorientierte

<sup>145</sup>Fähigkeiten algorithmischen Arbeitens weisen zudem enge Verbindungen zu heuristischen Strategien auf, die für die Mathematik und die Informatik gleichermaßen Relevanz besitzen, siehe S. 188.

<sup>146</sup>Das Verhältnis zwischen Klassen und Instanzen sowie die Anwendung von Methoden (geometrischen Transformationen) wird anhand dieses Beispiels deutlich. Allerdings implementierte der Schüler keine Attribute, die es ermöglichen würden, in den Instanzen des Schneemannes gewisse Eigenschaften wie z. B. Farben zu verändern. Alle erzeugten Instanzen sind somit bis auf ihre Lage und Größe, die durch geometrische Transformationen bestimmt werden, identisch.



Vorgehensweisen besonders deutlich hervortreten lässt.<sup>147</sup>

- Im Zusammenhang mit der strukturierten Gestaltung von Szenenbeschreibungen kommen in elementarer Weise *Aspekte modularen Arbeitens* zum Tragen.
- *Iterative Arbeitsweisen* kommen u. a. zur Anwendung, wenn Schüler Geraden oder Kurven durch große Anzahlen kleiner Kugeln darstellen und dazu *Schleifen* nutzen (siehe S. 152 und S. 237f.).
- Anhand von *Bézier- und Splinekurven* (vgl. 2.3.6) werden Bezüge zwischen Geometrie und Informatik gut deutlich. Auf eine ausführlichere Beschreibung dieses Themengebietes wurde jedoch aus den auf S. 289 genannten Gründen verzichtet.
- Mit dem *Raytracing-Verfahren* (siehe 2.5.5) kann ein interessanter *rekursiver Algorithmus* zumindest in seiner prinzipiellen Funktionsweise behandelt werden. Allerdings erscheint es wenig realistisch, dass Schüler im Mathematikunterricht selbst Raytracing-Algorithmen implementieren, so dass sich Überlegungen hierzu auf die Diskussion des Verfahrens, evtl. ergänzt um Experimente mit unterschiedlichen Strahlverfolgstiefen (siehe S. 355), beschränken dürften.

Die Tiefe und Ausführlichkeit, mit der auf informatische Aspekte anhand der aufgeführten Inhalte eingegangen werden kann, hängt von der zur Verfügung stehenden Zeit sowie davon ab, ob Schüler der betreffenden Klassen bzw. Kurse auch Informatikkurse belegen. Ist dies bei einem Teil der Schüler der Fall, so bietet es sich an, dass diese Schüler in stärkerem Maße z. B. iterative und rekursive Vorgehensweisen für die Erstellung von Computergrafiken und -animationen anwenden.<sup>148</sup>

Unter Berücksichtigung der Tatsache, dass Modellierung und Simulation zentrale Arbeitsweisen der Informatik darstellen, lassen sich noch an einer Reihe weiterer, z. B. in den Abschnitten 4.1, 4.3 und 4.6.5 beschriebener Gegenstände Anknüpfungspunkte zur Informatik ausmachen, wenngleich dabei mathematische Beschreibungen im Vordergrund stehen. Elemente der Computergrafik und -animation sind auch für den Informatikunterricht relevant,<sup>149</sup> wobei naturgemäß mathematische Aspekte einzubeziehen sind. Die Computergrafik stellt somit einen Gegenstandsbereich dar, anhand dessen reichhaltige Bezüge zwischen der Mathematik und der Informatik hergestellt werden können, wobei mitunter die Grenzen zwischen diesen (ohnehin nicht vollständig voneinander abgrenzbaren) Wissenschaften verschwimmen.

---

<sup>147</sup>Grundzüge objektorientierter Programmierung könnten z. B. bei Nutzung der objektorientierten Skriptsprache Python (siehe 2.6.5 und 2.7.2) besser herausgearbeitet werden. POV-Ray oder MuPAD wurde aus Gründen, die für die Nutzung im Stoffgebiet Analytische Geometrie gegenüber der Verdeutlichung des Konzepts der Objektorientierung vorrangig sind, der Vorzug gegeben, siehe 3.4.

<sup>148</sup>Differenzierungsmöglichkeiten bestehen u. a. dahin gehend, dass Schüler mit Informatikkenntnissen bei der Behandlung der Parameterdarstellungen von Geraden und Kurven Darstellungen unter Verwendung von Schleifen bzw. Prozeduren anfertigen, während die anderen Schüler nur einzelne Punkte darstellen (siehe S. 237f.), falls die vorhandene Zeit im Unterricht knapp bemessen ist.

<sup>149</sup>So beinhaltet der Hamburger Rahmenplan für das Wahlpflichtfach Informatik in den Klassenstufen 9 und 10 die Themengebiete *Grafik* (S. 10), *Bewegte Bilder* (S. 18) und *3D-Modellieren* (S. 19); siehe die Datei *Hamburg2003-Informatik-SI.pdf* in dem Ordner *Rahmenplaene* (siehe S. 365f.).

## 4.9 Varianten für den Aufbau des Stoffgebietes Analytische Geometrie unter Einbeziehung von Elementen der Computergrafik

Es wird wohl – allein schon aus Zeitgründen – in keinem Falle möglich sein, alle in diesem Kapitel beschriebenen Vorschläge für die Einbeziehung von Elementen der Computergrafik in den Unterricht des Stoffgebietes Analytische Geometrie in einem einzigen Kurs aufzugreifen. Im Folgenden sollen daher einige Curriculumsvarianten für dieses Stoffgebiet unter Einbeziehung der 3D-Computergrafik skizziert werden.

Eine unmittelbare Möglichkeit der Nutzung einer Visualisierungssoftware<sup>150</sup> besteht in der Veranschaulichung von Inhalten der in den meisten Rahmenplänen vorgesehenen vektoriellen Raumgeometrie (siehe 4.2), wobei die Software lediglich als Hilfsmittel genutzt wird. Abgesehen davon, dass es in Ergänzung zu der meist recht einseitig rechnerischen Behandlung von Vektoren, Geraden und Ebenen sinnvoll ist, wenn die Schüler eine Vorstellung von deren Richtung bzw. Lage im Raum erlangen, ergeben sich dadurch keine Veränderungen hinsichtlich der in Abschnitt 1.4 aufgeführten grundsätzlichen Probleme des Stoffgebietes, dessen Inhalte dabei nicht verändert werden. Eine derartige Nutzung von Visualisierungssoftware kann ohne weitere inhaltliche Überlegungen erfolgen und bedingt keine curricularen Konsequenzen.

Sollen Elemente der Computergrafik etwas umfassender, über die Nutzung als Werkzeug für die Visualisierung traditioneller Inhalte des Stoffgebietes hinausgehend, in das Stoffgebiet Analytische Geometrie einbezogen werden, so bildet der in Abschnitt 4.1 beschriebene Einstieg eine Basis sowohl für die Behandlung von Elementen der analytischen Geometrie des Raumes als auch für den Umgang mit einer im folgenden Unterrichtsverlauf verwendeten Software. Wie bereits ausgeführt wurde, ermöglicht ein derartiger Einstieg in das Stoffgebiet über die Beschreibung geometrischer Grundkörper durch Koordinaten auch ein Anknüpfen an Inhalte des Geometrieunterrichts der SI. Er bildet daher den Ausgangspunkt für alle im Folgenden skizzierten Lehrgangsvarianten.<sup>151</sup> Dabei ist einerseits zwischen stärker koordinatengeometrisch orientierten Lehrgängen (bis hin zu einem Verzicht auf die Behandlung von Vektoren) und vektoriell orientierten Lehrgängen sowie andererseits zwischen Basislehrgängen für Grundkurse und Lehrgangsvarianten für Leistungskurse zu unterscheiden. In Leistungskursen

<sup>150</sup>Der Begriff „Visualisierungssoftware“ wird hier verallgemeinernd für Software, mit deren Hilfe die Darstellung von Objekten der räumlichen analytischen Geometrie möglich ist, verwendet. Sollen lediglich Inhalte der vektoriellen Raumgeometrie veranschaulicht werden, erscheint die Nutzung einer fotorealistischen Grafiksoftware wie POV-Ray nicht sinnvoll. Die Argumente, die in Abschnitt 3.4 für deren Nutzung genannt wurden, kommen nicht zum Tragen, wenn lediglich Vektoren, Geraden und Ebenen sowie ihre Lagebeziehungen veranschaulicht werden. In diesem Falle ist die Nutzung eines CAS oder einer speziell für den Unterricht in analytischer Geometrie entwickelten Software (siehe 2.7.6) angebrachter, deren Bedienung mit sehr geringem Einarbeitungsaufwand möglich ist.

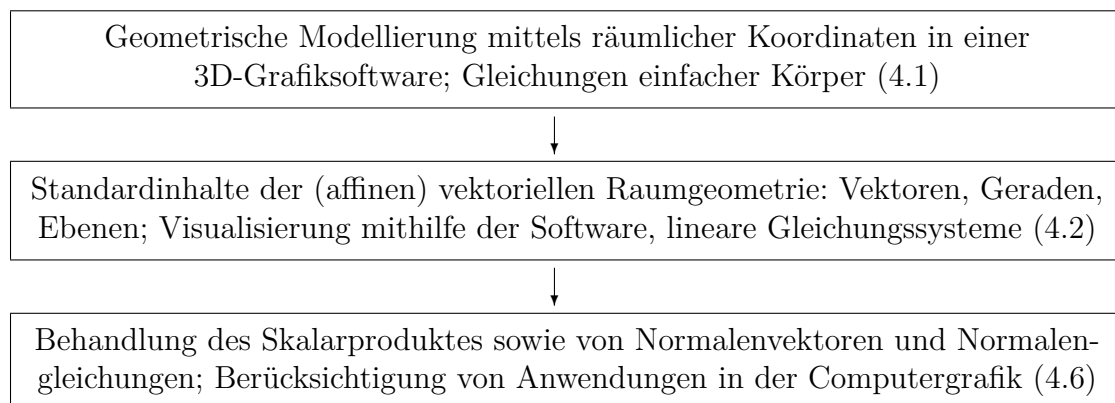
<sup>151</sup>Gründe dafür, das Stoffgebiet Analytische Geometrie mit auf Koordinatenbeschreibungen basierenden Modellierungen geometrischer Objekte unter Verwendung einer Grafiksoftware bzw. eines CAS zu beginnen, sowie damit verfolgte Ziele wurden bereits auf S. 217 genannt.

erscheint u. a. eine stärkere Berücksichtigung anspruchsvoller Querverbindungen zur Analysis besonders wünschenswert.<sup>152</sup>

#### 4.9.1 Lehrgangsvarianten für Grundkurse

##### Einbeziehung von Elementen der 3D-Computergrafik in einen Lehrgang, dessen Schwerpunkt die vektorielle Raumgeometrie darstellt

Wie aus Abschnitt 1.3 hervorgeht, bildet die vektorielle Behandlung von Geraden und Ebenen den Schwerpunkt des Stoffgebietes Analytische Geometrie nach den Rahmenplänen der meisten Bundesländer. Oft wird dieser Schwerpunkt also auch zu berücksichtigen sein, wenn Elemente der Computergrafik in den Unterricht einbezogen werden. Dem kann mit dem folgenden Aufbau des Stoffgebietes Rechnung getragen werden:<sup>153</sup>



Wie bereits auf S. 232 erwähnt wurde, stellt der Übergang von koordinatengeometrischen zu vektoriellen Vorgehensweisen bei dem skizzierten Lehrgangsaufbau einen inhaltlichen Bruch dar. Die Einführung von Vektoren ausgehend von einer räumlichen Koordinatengeometrie zu motivieren, würde eine wesentlich weiter gehende Beschäftigung mit koordinatengeometrischen Vorgehensweisen erfordern, um zu deren Grenzen bzw. zu Inhalten zu gelangen, anhand derer die Einführung von Vektoren überzeugende und nachvollziehbare Vorteile gegenüber Koordinatenbeschreibungen mit sich bringt (siehe z. B. [73], S. 96f.). Bei dem hier beschriebenen Aufbau stört zudem, dass den zwar elementaren, jedoch geometrisch interessanten Überlegungen zu Kreisen und Kugeln die ausführliche Beschäftigung mit Geraden und Ebenen folgt, was in Bezug auf die behandelten geometrischen Objekte als „Rückschritt“ aufgefasst werden kann.

<sup>152</sup>In jedem Falle sollten Möglichkeiten der inneren Differenzierung berücksichtigt werden. Bei der Einbeziehung von Elementen der Computergrafik sind diese von besonderer Bedeutung, da entsprechende Themen auch für viele Schüler in Grundkursen attraktiv sind und aus diesem Grund Bereitschaft zu erwarten ist, über die im Unterricht behandelten Inhalte hinaus mit der Erstellung von Computergrafiken und -animationen verbundene Aufgaben zu bearbeiten. Besonderes Augenmerk ist auf Möglichkeiten der inneren Differenzierung zudem in Bundesländern wie Baden-Württemberg zu legen, in denen im Fach Mathematik keine Differenzierung nach Grund- und Leistungskursen mehr erfolgt.

<sup>153</sup>Die Ziffern in Klammern verweisen auf die Abschnitte, in denen auf die Behandlung der entsprechenden Inhalte eingegangen wurde.

Trotz der genannten Probleme stellt der auf S. 293 skizzierte Aufbau des Stoffgebietes einen Kompromiss dar, der den Vorgaben vieler Rahmenpläne Rechnung trägt und dabei dennoch attraktive Anwendungen der 3D-Computergrafik einbezieht und deren Motivierungspotenzial zumindest im ersten und dritten Teil des Stoffgebietes nutzt. Zudem ist die Behandlung von Kreisen und Kugeln im ersten Lehrgangsteil zumindest ein Ansatz, die durch die Beschränkung auf Geraden und Ebenen bedingte Formenarmut des gegenwärtigen Unterrichts in analytischer Geometrie (vgl. 1.4.5) etwas zu lindern und geometrische Fragen stärker in den Mittelpunkt zu stellen.<sup>154</sup> Kleinere curriculare Veränderungen könnten zudem den nach den gegenwärtigen Rahmenplänen recht hohen Zeitaufwand für die algebraisch-kalkülmäßige Lösung von Standardaufgaben (wie Schnittmengenbestimmungen durch das Lösen von Gleichungssystemen, Abstands- und Winkelberechnungen) verringern<sup>155</sup> und damit Freiräume für Modellierungstätigkeiten schaffen. Diese Freiräume ließen sich u. a. für die parameterabhängige Beschreibung von Bahnkurven und die darauf basierende Erstellung von Animationen (vgl. 4.3) nutzen.<sup>156</sup> Eine weitere, mit recht geringem Zeitaufwand mögliche, Ergänzung bei der Behandlung von Vektoren mit Blick auf Anwendungen in der Computergrafik stellt die Beschreibung von Farben durch Vektoren dar, siehe 2.1.2 und 3.3.1.

### **Räumliche Koordinatengeometrie als Schwerpunkt des Stoffgebietes Analytische Geometrie**

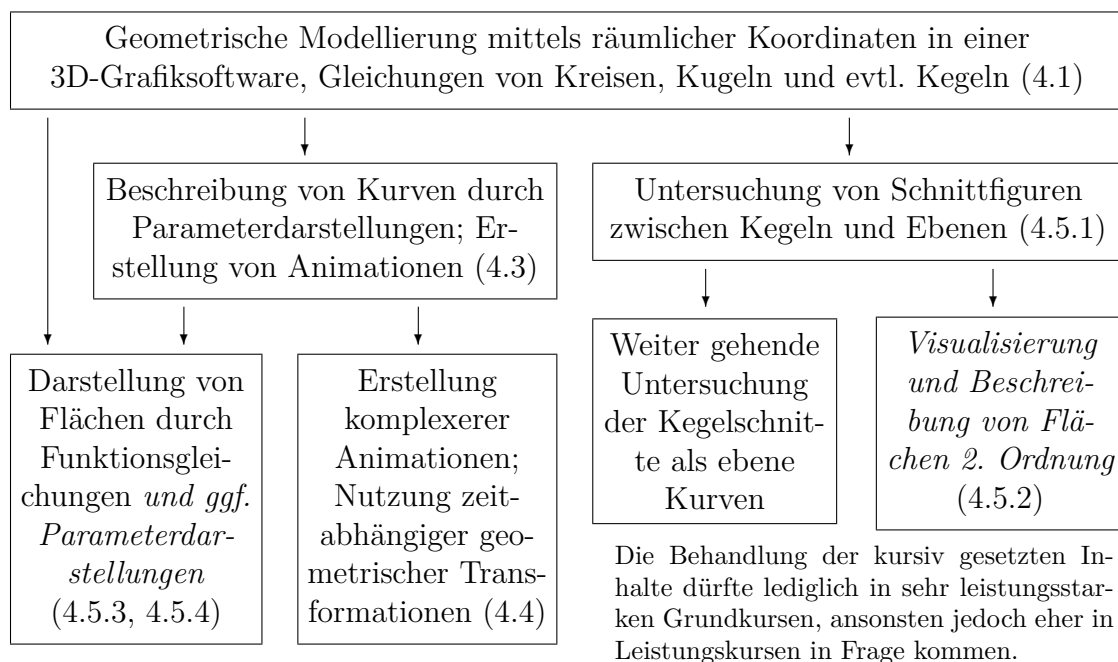
Obwohl die Vektorrechnung gegenwärtig zu den Kerninhalten des Mathematikunterrichts der Sekundarstufe II zählt, sind die dabei erreichten Ergebnisse sowohl hinsichtlich der Anwendungsbreite als auch in Bezug auf die Tiefe des bei den Schülern erreichten Verständnisses des Vektorbegriffs (siehe u. a. [276]) insbesondere in Grundkursen oftmals unbefriedigend und lassen den für die Behandlung des Vektorkalküls notwendigen Zeitaufwand fragwürdig erscheinen. Insofern sollte der Verzicht auf die Behandlung von Vektoren bei curricularen Überlegungen in Erwägung gezogen werden (siehe auch Fuß-

<sup>154</sup>Aufgrund der Kompatibilität zu den geltenden Rahmenplänen wurden die in 5.1 und 5.2 beschriebenen Unterrichtsversuche auf der Grundlage der auf S. 293 dargestellten Grobgliederung durchgeführt, wobei in einem dieser Unterrichtsversuche aus Zeitgründen auf die Behandlung der Kreis- und Kugelgleichungen verzichtet werden musste. Trotz der genannten Probleme begrüßten die Schüler – wie in Kapitel 5 noch ausführlicher dargestellt wird – die Einbeziehung von Elementen der Computergrafik aufgrund der dadurch erreichten Anschaulichkeit und Praxisrelevanz des Unterrichts. Somit bringt auch diese Variante gegenüber der vielfach recht einseitig auf die Durchführung von Berechnungen konzentrierten Unterrichtspraxis im Stoffgebiet Analytische Geometrie Verbesserungen mit sich.

<sup>155</sup>In der gegenwärtig häufig anzutreffenden Unterrichtspraxis wird ein sehr großer Teil der Unterrichtszeit für die Lösung dieser Routineaufgaben verwendet (vgl. 1.4.2). Die Reduktion dieses Zeiteils muss daher den wichtigsten Ansatzpunkt darstellen, um interessantere Inhalte und Anwendungen im Stoffgebiet Analytische Geometrie zu berücksichtigen. Hierzu kann, neben der Verringerung der zu behandelnden Aufgabentypen, die Nutzung von CAS bzw. geeigneten grafischen Taschenrechnern für die Lösung linearer Gleichungssysteme beitragen.

<sup>156</sup>Da die Erstellung von Animationen für viele Schüler sehr reizvoll ist, kann diese Thematik allerdings auch unter Bedingungen erheblicher Zeitknappheit im Unterricht zumindest „angerissen“ werden, wobei nach den in Unterrichtsversuchen gesammelten Erfahrungen die Wahrscheinlichkeit hoch ist, dass zumindest einzelne Schüler in ihrer Freizeit Animationen generieren werden.

note 39 auf S. 197). Insbesondere anhand der Abschnitte 4.3 und 4.5 wird deutlich, dass vielfältige, geometrisch interessante und gleichzeitig mathematisch anspruchsvolle Überlegungen und Anwendungen unter Einbeziehung von Elementen der 3D-Computergrafik auf der Grundlage räumlicher Koordinatengeometrie möglich sind. Wie bereits in Abschnitt 4.1.8 auf S. 231 ausgeführt wurde, können die meisten der in 4.3 und 4.5 beschriebenen Themengebiete direkt auf der Grundlage des in 4.1 beschriebenen Einstiegs in das Stoffgebiet Analytische Geometrie behandelt werden. Das folgende Schema zeigt einige Möglichkeiten, ein Stoffgebiet zur räumlichen Koordinatengeometrie unter intensiver Nutzung dreidimensionaler Visualisierungen aufzubauen.



Wie weit bei der Behandlung der einzelnen Themengebiete gegangen werden kann, hängt natürlich stark von der zur Verfügung stehenden Zeit, aber auch von den Voraussetzungen der Schüler ab.<sup>157</sup> Alle in dem oben abgebildeten Schema enthaltenen Themengebiete im Unterricht aufzugreifen, erscheint unrealistisch; vielmehr wird sich die Konzentration auf einen der angegebenen „Stränge“ als notwendig erweisen.<sup>158</sup>

<sup>157</sup>Siehe dazu die Bemerkung auf S. 224 – falls die Schüler zu Beginn des Stoffgebietes z. B. noch nicht in der Lage sind, den Satz des Pythagoras für die Berechnung des Abstandes zweier durch Koordinaten gegebener Punkte im Raum zu nutzen, so sollte einer ausführlichen Behandlung der in Abschnitt 4.1 beschriebenen Inhalte der Vorzug gegenüber einer Bearbeitung möglichst vieler Themengebiete gegeben werden, hinsichtlich derer die Schüler dann weder ein tieferes Verständnis noch die Fähigkeit der flexiblen Anwendung in unterschiedlichen Kontexten erlangen. Meist dürften die Voraussetzungen bei den Schülern recht unterschiedlich sein. Wie bereits ausgeführt wurde, bieten vor allem die mit Animationen im Zusammenhang stehenden Themenkreise gute Möglichkeiten innerer Differenzierung.

<sup>158</sup>Dies bedeutet nicht, dass auf alle anderen Themengebiete vollständig zu verzichten wäre. So wird die elementare Beschreibung geometrischer Transformationen (vgl. 4.4.3) z. B. für Positionierungen in aus mehreren Objekten zusammengesetzten Szenen benötigt und kann in diesem Zusammenhang „zweckbezogen“ angesprochen werden, ohne sie gründlicher zu thematisieren.

Da zwar die in Abschnitt 4.1 beschriebenen Inhalte die Grundlage für alle weiteren Betrachtungen bilden, diese dann jedoch in weiten Teilen unabhängig voneinander geführt werden können,<sup>159</sup> bieten sich gute Möglichkeiten für einen *projektorientierten Unterricht*, bei dem Schüler selbst Themenschwerpunkte wählen können, mit denen sie sich vertiefend beschäftigen. Denkbar sind dabei auch Kooperationen, die zu gemeinsamen Ergebnissen führen. So können Schüler, die sich schwerpunktmäßig mit Animationen beschäftigen und solche, die sich für anspruchsvollere Aspekte der geometrischen Modellierung (z. B. die Darstellung von Flächen)<sup>160</sup> entschieden haben, ihre Arbeitsergebnisse zu Videos zusammenfügen, die interessante geometrische Formen in anspruchsvollen Animationen aus unterschiedlichen Perspektiven darstellen.<sup>161</sup>

#### 4.9.2 Einbeziehung von Elementen der 3D-Computergrafik in den Unterricht von Leistungskursen

Die im vergangenen Abschnitt für Grundkurse beschriebenen Varianten für die Einbeziehung von Elementen der 3D-Computergrafik in Kurse mit dem Schwerpunkt einer vektoriellen Raumgeometrie sowie für die Behandlung anspruchsvoller geometrischer Inhalte auf Koordinatenbasis (siehe die Schemata auf S. 293 und S. 295) eignen sich auch als Ausgangspunkte für entsprechende Überlegungen zur Gestaltung von Leistungskursen.<sup>162</sup> Gegenüber Grundkursen sind jedoch in Leistungskursen einige Spezifika hinsichtlich der Herangehensweisen und der Schwerpunkte bei der Behandlung der

<sup>159</sup>Ausnahmen werden aus dem Schema auf S. 295 ersichtlich. So sollten Parameterdarstellungen von Flächen (die ein vor allem für Leistungskurse geeignetes Themengebiet darstellen) nicht behandelt werden, bevor Kurven durch Parameterdarstellungen beschrieben wurden. Der Behandlung von Flächen zweiter Ordnung sollte die (zumindest elementare) Untersuchung der Kegelschnitte vorausgehen.

<sup>160</sup>Weitere Möglichkeiten der geometrischen Modellierung ergeben sich u. a. durch die Anwendung Boolescher Operationen (vgl. 2.3.4) sowie durch die Beschreibung von Rotationsflächen bzw. -körpern, wobei Zugänge mit unterschiedlichen Schwierigkeitsgraden der zugrunde liegenden mathematischen Beschreibungen in Frage kommen (vgl. 2.3.7 und 4.5.4).

<sup>161</sup>Bei ausreichend vorhandener Zeit käme als dritte Komponente die Gestaltung von Objektoberflächen (Texturierung) auf Grundlage der in 4.6.5 beschriebenen Überlegungen in Frage. Dies würde jedoch das Themenspektrum erheblich erweitern und vektorielle Betrachtungen erfordern. Zudem erscheinen die den Beleuchtungsmodellen zugrunde liegenden mathematischen Grundlagen (siehe 2.5.2 und 4.6) im Gegensatz zu den Grundlagen von Animationen weniger geeignet, von den Schülern weitgehend selbstständig erarbeitet zu werden. Projekte, welche Modellierung, Animation und Texturierung auf der Grundlage mathematischer Modelle beinhalten, kommen deshalb nur dann in Frage, wenn innerhalb eines Lehrgangs sowohl anspruchsvolle koordinatengeometrische als auch vektorielle Inhalte behandelt werden können, was jedoch gerade in Grundkursen eine seltene Ausnahme darstellen dürfte.

<sup>162</sup>Die meisten Rahmenpläne sehen für das Stoffgebiet Analytische Geometrie keine grundsätzlichen inhaltlichen Unterschiede zwischen Grund- und Leistungskursen vor. Die Leistungskurscurricula unterscheiden sich von denen der Grundkurse vor allem durch zusätzlich zu führende strukturelle Betrachtungen (vgl. 1.3). Auch aus diesem Grunde werden Grundkurscurricula häufig als „ausgedünnte Leistungskurse“ kritisiert, siehe u. a. [29], S. 79 – die mit der geometrischen Reduktion und der Unterordnung der analytischen Geometrie unter die lineare Algebra ursprünglich intendierten strukturellen Betrachtungen sind in Leistungskursen zumindest ansatzweise realisierbar, während sich in Grundkursen das Stoffgebiet hingegen in vielen Fällen weitgehend auf das Abarbeiten von Kalkülen reduziert.

genannten Themengebiete zu beachten; zugleich eröffnen sich zusätzliche Möglichkeiten.

- Die Motivation durch außermathematische Anwendungen ist für Leistungskurse weniger relevant als für Grundkurse, da Schüler in Leistungskursen naturgemäß stärker durch innermathematische Problemstellungen motiviert werden können. Der für Grundkurse immens bedeutsame Motivierungsaspekt durch Anwendungen der behandelten mathematischen Inhalte in der 3D-Computergrafik hat dadurch in Leistungskursen ein etwas geringeres Gewicht. Allerdings können erfahrungsgemäß praxisrelevante computergrafische Anwendungen und die Ästhetik selbst erstellter Grafiken sowie die Anfertigung von Animationen auch mathematisch interessierte und talentierte Schüler motivieren.
- Das in Leistungskursen höhere Maß an verfügbarer Zeit ermöglicht es in vielen Fällen, sowohl vektorielle Methoden zu behandeln als auch anspruchsvollere koordinatengeometrische Beschreibungen zu berücksichtigen.<sup>163</sup>
- Die anwendungsorientierte Behandlung von Inhalten der analytischen Geometrie und Modellierungsprozesse realer Sachverhalte sind sowohl in Grund- als auch in Leistungskursen angemessen, wobei in Letzteren anspruchsvollere theoretische Betrachtungen einzubeziehen sind.<sup>164</sup>

In Leistungskursen zur analytischen Geometrie bieten sich vor allem zwei Richtungen hinsichtlich theoretischer Vertiefungen und Exaktifizierungen an:

- Bei einem schwerpunktmäßig auf die vektorielle Raumgeometrie orientierten Lehrgang lassen sich strukturelle Fragen der linearen Algebra stärker berücksichtigen als in Grundkursen.<sup>165</sup> Möglich erscheint u. a. die Behandlung von Matrizen.<sup>166</sup>
- Auch bei einem koordinatengeometrisch orientierten Aufbau des Stoffgebietes Analytische Geometrie ergeben sich sehr interessante Möglichkeiten anspruchsvoller mathematischer Überlegungen und Modellierungen. Dazu gehört vor allem die Behandlung von Raumkurven und Flächen (vgl. 4.3 und 4.5), die zwar auch in Grundkursen ansatzweise möglich erscheint, in Leistungskursen jedoch theoretisch vertieft und insbesondere um weiter gehende Bezüge zur Analysis ergänzt werden kann.

---

<sup>163</sup>Eine Schwerpunktsetzung zugunsten von Inhalten der vektoriellen analytischen Geometrie und der linearen Algebra oder für die Untersuchung von Kurven und Flächen mit Methoden der Koordinatengeometrie unter Herstellung von Bezügen zur Analysis wird dennoch erforderlich sein.

<sup>164</sup>Dass sich im Zusammenhang mit Inhalten der Computergrafik aus mathematischer Sicht sehr anspruchsvolle Betrachtungen ergeben können, wird an einigen in Kapitel 2 beschriebenen Beispielen und auch anhand der in den vorangegangenen Abschnitten unterbreiteten Vorschläge deutlich.

<sup>165</sup>Einige für Leistungskurse spezifische Themengebiete führte TIETZE in [247], S. 157f. auf. Er wandte sich dabei allerdings gegen die axiomatische Behandlung von Vektorräumen. Hierfür sowie z. B. für axiomatisch-strukturelle Varianten der Einführung des Skalarproduktes ist auch bei Schülern in Leistungskursen i. Allg. keine hinreichende Abstraktionsbasis gegeben.

<sup>166</sup>Auch hierbei lassen sich interessante Bezüge zur Computergrafik herstellen, siehe Abschnitt 2.4.

### Bezüge zum Unterricht in Analysis

Auch in Leistungskursen erscheinen die Stoffgebiete Analysis sowie Analytische Geometrie / Lineare Algebra nach den gegenwärtig verbreiteten Curricula für die Schüler weitgehend voneinander isoliert (siehe S. 52). Eine stärkere Berücksichtigung koordinatengeometrischer Vorgehensweisen und auf dieser Grundlage erfolgende Untersuchungen von Kurven und Flächen ermöglichen es, diese Isolation aufzuheben und Vernetzungen zwischen beiden Themengebieten herzustellen.

Elementare Bezüge zwischen der analytischen Geometrie und der Analysis werden bereits deutlich, wenn Kurven und Flächen in der analytischen Geometrie untersucht und dabei, wie u. a. in Abschnitt 4.5.3 beschrieben, die Schnitte mit Koordinatenebenen betrachtet werden.<sup>167</sup> Wie bereits ausgeführt wurde, erleichtern computergestützte Visualisierungen derartige Betrachtungen erheblich und ermöglichen es, davon ausgehend Formen zu modellieren. Allerdings bilden dabei zwar Funktionsterme und -graphen ein Bindeglied zwischen beiden Stoffgebieten, die Nutzung von Elementen der Differenzial- und Integralrechnung erfolgt jedoch nicht. Deren Integration in das Stoffgebiet Analytische Geometrie und damit eine weiter gehende Vernetzung ist u. a. bei der Behandlung von Tangenten und Tangentialebenen möglich (siehe z. B. [66] und [279]).

Auf zwei für die Einbeziehung von Elementen der Computergrafik und -animation in das Stoffgebiet Analytische Geometrie spezifische Möglichkeiten, Verbindungen zur Differenzialrechnung herstellen, wurde in der vorliegenden Arbeit bereits kurz eingegangen:

- Bei der Darstellung von Flächen, die durch implizite Gleichungen gegeben sind, führt die Software POV-Ray Intervallunterteilungen durch. Für die Vollständigkeit der Flächendarstellung und die dazu benötigte Zeit ist ein vorzugebender Wert für den maximalen Gradienten entscheidend. Dessen Bedeutung erschließt sich durch die Reduktion auf Funktionen einer Veränderlichen und die Anwendung des Mittelwertsatzes der Differenzialrechnung (siehe S. 93ff.). Jedoch sind diese Überlegungen anhand von Anwendungen nur dann zu rechtfertigen, wenn zuvor Flächen recht ausführlich untersucht und in POV-Ray dargestellt wurden.
- Bei der Erstellung von Animationen auf Bahnkurven lässt sich die Bahngeschwindigkeit von Bewegungen und im Zusammenhang damit die Bogenlänge thematisieren (siehe Abschnitt 4.3.3).

Diese beiden Themengebiete befinden sich auf einem für Schulverhältnisse sehr hohen mathematischen Niveau, das sich ohne deutlichen Substanzverlust nicht weiter reduzieren lässt, als dies in den Darstellungen auf S. 93ff. bzw. auf S. 254 bereits erfolgt ist. Ein Aufgreifen dieser Themen dürfte daher nur in besonders leistungsfähigen Leistungskursen bzw. im Rahmen innerer Differenzierung für einzelne Schüler in Frage kommen.

<sup>167</sup>Elementare Bezüge zwischen den beiden Stoffgebieten, zumindest hinsichtlich der untersuchten Objekte, ergeben sich auch durch die Modellierung von Rotationskörpern im Stoffgebiet Analytische Geometrie, wenn der Berechnung von deren Volumina bei der Behandlung der Integralrechnung eine wichtige Bedeutung zukommt. Dieser Bezug war einer der Schwerpunkte eines von F. RIEPER durchgeführten Unterrichtsprojektes in einem Grundkurs, auf das in Abschnitt 5.3 eingegangen wird.



## Kapitel 5

# Unterrichtserfahrungen hinsichtlich der Einbeziehung von Elementen der 3D-Computergrafik

In Unterrichtsreihen am Andreas-Gymnasium in Berlin und an der Internationalen Gesamtschule Heidelberg wurden Elemente der Computergrafik punktuell in das Stoffgebiet Analytische Geometrie einbezogen. Die dabei gesammelten Erfahrungen sollen einen Einblick in die nach zwei gegenwärtig gültigen Rahmenplänen bestehenden Möglichkeiten geben, Potenziale der Arbeit mit computergrafischen Darstellungen im Unterricht zu nutzen sowie in Ansätzen Grundlagen der Computergrafik zu thematisieren. Insbesondere wird dargelegt, wie Schüler mit einer Grafiksoftware arbeiteten und wie sie die Einbeziehung von computergrafischen Inhalten in den Unterricht bewerteten.<sup>1</sup>

Etwas umfangreichere Freiräume für die Arbeit von Schülern an 3D-Computergrafiken bestanden in einem dreiwöchigen Unterrichtsprojekt in einem Mathematik-Grundkurs am Fürst-Pückler-Gymnasium in Cottbus, auf das in Abschnitt 5.3 kurz eingegangen wird. Ergänzend erfolgt eine Schilderung von Erfahrungen aus der Arbeit mit Studierenden in Seminaren zur geometrischen Modellierung in der Computergrafik.

Nicht alle in Kapitel 4 unterbreiteten Vorschläge konnten in Unterrichtsversuchen erprobt werden. Die Richtungen einiger Unterrichtsverläufe und die gesammelten Erfahrungen erlauben jedoch gewisse Aussagen darüber, welche Themengebiete Erfolg versprechend in den Unterricht einbezogen werden können.

Als Ergänzung der im Folgenden beschriebenen Unterrichtserfahrungen enthält der Ordner **Schuelerarbeiten\Facharbeit-S-Haaf** die Facharbeit eines Schülers, der sich mit mathematischen Grundlagen der Computergrafik und dabei insbesondere mit Koordinatentransformationen von Szenen- in Ansichtskoordinatensysteme beschäftigte. Aus Platzgründen wird auf diese Einzelarbeit nicht näher eingegangen; Anmerkungen zu ihrer Entstehung finden sich in dem genannten Ordner (siehe Anhang D, S. 365).

---

<sup>1</sup>Die beschriebenen Unterrichtsversuche und Schülerbefragungen können schon aufgrund ihres relativ geringen Umfangs nicht als fundierte empirische Untersuchungen angesehen werden; es erfolgt lediglich eine Darlegung und qualitative Auswertung exemplarisch gesammelter Erfahrungen.

## 5.1 Unterrichtsversuch in einem Grundkurs ma-3, Andreas-Gymnasium, Berlin

In einem Grundkurs ma-3 (1. Halbjahr des 13. Schuljahres) an der Andreas-Oberschule (Gymnasium) in Berlin-Friedrichshain wurde vom 20. August bis zum 1. Dezember 2003 das Stoffgebiet Analytische Geometrie unter Einbeziehung von Elementen der 3D-Computergrafik unterrichtet. Den Unterricht führte der Autor der vorliegenden Arbeit in Kooperation mit Herrn BECHMANN, dem Lehrer des betreffenden Kurses, durch. Dem Kurs gehörten 12 Schülerinnen und 8 Schüler an.

### 5.1.1 Rahmenbedingungen, Voraussetzungen der Schüler

Der Unterricht erfolgte nach den in Abschnitt 1.3 ab S. 26 wiedergegebenen Rahmenplänen des Landes Berlin, wobei die für die Behandlung der analytischen Geometrie gesetzte Zeitvorgabe von 30 Unterrichtsstunden mit geringen Abweichungen einzuhalten war. In der 11. Klasse sowie in den Kurshalbjahren ma-1 und ma-2 wurde der Unterricht von anderen Lehrern durchgeführt. Es konnte in Erfahrung gebracht werden, dass die für das Stoffgebiet Lineare Gleichungssysteme im Kurs ma-2 vorgesehene Zeit von 15 Unterrichtsstunden am Ende von Klasse 12 nicht vollständig zur Verfügung stand und dieses Themengebiet daher nur ansatzweise behandelt wurde. Um einen ersten groben Überblick über die vor allem im Unterricht der Klasse 11 erworbenen Vorkenntnisse der Schüler zu erhalten, erfolgte in der ersten Unterrichtsstunde eine kurze (anonyme) Befragung, wobei die Schüler zwei Fragen beantworten sollten:

1. Was fällt Ihnen zu dem Begriff *Vektor* ein – was ist ein Vektor, kennen Sie Beispiele?
2. Ein Punkt hat die Koordinaten  $(2; -3; 4)$ . Was sagt dies über die Lage des Punktes im Raum aus? Beantworten Sie die Frage mit Worten und ergänzen Sie Ihre Antwort evtl. durch eine grobe Skizze.

An der Befragung nahmen 18 Schüler teil; auf die Frage 1 gaben sie folgende Antworten:<sup>2</sup>

- Ein Vektor gibt die Richtung einer Kraft an.
- Ein Vektor ist (glaube ich) eine Art Zeiger. Jedenfalls verläuft ein Vektor nur in eine Richtung im Raum.
- $\vec{v}$  Ich habe leider keine Ahnung mehr!
- Vektoren sind mir vor allem aus dem Physikunterricht bekannt. Meistens beschreiben wir damit Kräfte, die auf ein bestimmtes Objekt wirken. Sie geben deren Richtung und Stärke an. Man kann ein Parallelogramm zeichnen, um eine resultierende Kraft zu ermitteln.
- Vektorenrechnung benutzt man neben d. Mathematik auch in Physik. Ein Vektor bzw. Vektoren bezeichnen die Richtung, in die eine Kraft wirkt (Physik).

---

<sup>2</sup>Die Schülerantworten sind hier wörtlich wiedergegeben; Korrekturen an der Orthographie und der Interpunktion wurden nicht vorgenommen.

- Ein Vektor ist die Menge aller Pfeile, die gleichlang, zueinander parallel und gleichgerichtet sind  $\hookrightarrow$  Pfeilklassse. Begriff des Vektors!!
- Beispiel: Zur Berechnung von Kräften in der Physik. Gibt Verbindung zwischen zwei Punkten im Raum eine Richtung (Ortsvektor)
- –
- Ein Vektor bezeichnet eine Richtung. Beispiele kenne ich keine.
- in der Physik ist eine Kraft / es ist 1 Punkt in der Mathematik, der 3D im Koordinatensystem dargestellt wird und aus 3 Variablen besteht
- Vektoren in Physik geben Richtung einer Kraft an. Vektoren in Mathematik kann man addieren, dividieren und so. werden durch drei Punkte bestimmt. Können auch mehrdimensional betrachtet werden.
- Ein Vektor ist sozusagen ein Punkt im Raum, der durch drei Koordinaten im dreidimensionalen Koordinatensystem beschrieben wird.
- Vektor ist eine ~~Größe~~ Punkt im Raum. Dieser besitzt bestimmte Koordinaten! Diese zusammen ergeben die Lage im Raum.
- Vektor ist ein Begriff für einen Pfeil. Es gibt auch die Vektorrechnung.
- Zu dem Begriff an sich fällt mir nur Richtung und Pfeil ein.
- Ein Vektor beschreibt einen Punkt im Raum. Vektoren beschreiben meist eine Richtung.  $\mapsto A = \begin{pmatrix} 1 \\ 3 \\ -4 \end{pmatrix}$
- Richtung;  $\mapsto$  ; Kraft
- 1. Vektorrechnung
- 2.
- 3. Beispiele: Kraft, Geschwindigkeit

Die Antworten verdeutlichen, dass die Schüler zu Beginn des Kurses nur über sehr vage Vorstellungen hinsichtlich des Vektorbegriffs verfügten, die zum großen Teil im Physikunterricht entstanden waren (siehe auch S. 198). An zentrale Inhalte des nach dem Rahmenplan vorgesehenen Stoffgebietes „Analytische Geometrie“ in Klasse 11 (vgl. S. 26f.) konnte sich die Mehrheit der Schüler offensichtlich kaum erinnern.<sup>3</sup> Bei Betrachtung einiger Antworten ist zu vermuten, dass die 2. Frage (nach der Lage eines durch Koordinaten gegebenen Punktes im Raum) Schüler an die Koordinatendarstellung von Vektoren erinnerte und somit auch deren Antworten auf die 1. Frage beeinflusste.

---

<sup>3</sup>Es konnten keine Informationen über den Verlauf des Stoffgebietes Analytische Geometrie in den 11. Klassen, denen die Schüler des Kurses angehörten, in Erfahrung gebracht werden. Erfahrungsgemäß kam es häufig vor, dass die geplanten 30 Unterrichtsstunden für das Stoffgebiet „Einführung in die Analytische Geometrie“ am Ende von Klasse 11 nicht einmal annähernd zur Verfügung standen (siehe Abschnitt 1.3). Ursachen bestanden in Schwierigkeiten und daraus resultierenden Verzögerungen bei der (vorher erfolgenden) Behandlung der Differenzialrechnung sowie relativ häufigen Stundenausfällen am Ende des Schuljahres.

Aus den Antworten auf die auf S. 300 genannte Frage 2 geht hervor, dass ca. die Hälfte der Schüler mit den Koordinaten eines Punktes im Raum gewisse anschauliche Vorstellungen verband. Viele Schüler konnten die Frage jedoch nicht beantworten und auch keine Skizze anfertigen, aus der die Lage des Punktes annähernd zu „erraten“ wäre.<sup>4</sup>

Für die Planung der Unterrichtsreihe ergab sich aus der Eingangsbefragung und einigen Gesprächen mit Schülern, dass ein Aufbauen auf zuvor behandelte und bei den Schülern sicher verfügbare Inhalte der analytischen Geometrie nicht möglich sein dürfte und das Stoffgebiet dadurch eine „Einführung in die analytische Geometrie“ enthalten muss. Dies brachte natürlich mit sich, dass die Zeitvorgabe von 30 Stunden in Verbindung mit den Rahmenplanvorgaben nur recht geringe Spielräume lies. Da eine vergleichbare Situation erfahrungsgemäß recht häufig gegeben sein dürfte, erschien es dennoch lohnenswert, zu prüfen, in welchem Umfang die Einbeziehung von Elementen der Computergrafik möglich und sinnvoll ist.

### 5.1.2 Planung und Verlauf der Unterrichtsreihe

Aufgrund der beschriebenen zeitlichen Bedingungen konnten bei weitem nicht alle der in Kapitel 4 beschriebenen Möglichkeiten der Einbeziehung von Elementen der 3D-Computergrafik aufgegriffen werden. Vielmehr musste sich die Auswahl an den Rahmenplanvorgaben für das Stoffgebiet Analytische Geometrie im Grundkurs ma-3 nach den alten Berliner Rahmenplänen (siehe S. 28ff.) orientieren. Da die notwendigen Voraussetzungen aus Klasse 11 nicht in ausreichendem Maße gegeben waren (wie anhand der Ergebnisse des Eingangstests deutlich wurde), erschien es notwendig, die Beschreibung von Punkten durch Koordinaten im Raum sowie den Vektorbegriff etwas ausführlicher zu thematisieren, als dies für den Kurs ma-3 vorgesehen war. Zudem ist, wie in Abschnitt 4.1 ausgeführt wurde, ein Einstieg in das Stoffgebiet über die Arbeit mit räumlichen Koordinaten und die Beschreibung einfacher Körper durch Koordinaten bestimmender Punkte sowie charakteristische Größen essenziell für die Einbeziehung von Elementen der 3D-Computergrafik in das Stoffgebiet Analytische Geometrie. Es wurde daher folgende Grobgliederung des Stoffgebietes vorgenommen:

1. Arbeit mit räumlichen Koordinaten, Modellierung mithilfe einfacher geometrischer Objekte, Einarbeitung in die Software POV-Ray (siehe Abschnitt 4.1);
2. Einführung von Vektoren, Parameterdarstellungen von Geraden und Ebenen, Lagebeziehungen; Koordinatenform der Ebenengleichung; Nutzung von POV-Ray für die Veranschaulichung dieser Inhalte (siehe 4.2);
3. Skalarprodukt und Normalenvektoren unter Berücksichtigung von Anwendungen in der Computergrafik (siehe 4.6).

---

<sup>4</sup>Da die Antworten auf die Frage 2 (siehe S. 300) nur im Zusammenhang mit den von den Schülern angefertigten Skizzen einzuordnen sind, wird auf einen Abdruck hier verzichtet. Kopien aller von den Schülern ausgefüllten Fragebögen befinden sich in der elektronischen Materialsammlung zu der vorliegenden Arbeit unter <http://www.afiller.de/habil> (Ordner **Schulversuch-Andreas-Gym**, Datei **Eingangstest-Antworten-Scans.pdf**; siehe Anhang D, S. 365).

### Der Einstieg in das Stoffgebiet

Im Folgenden wird stichpunktartig der Verlauf des ersten Teils der Unterrichtsreihe skizziert; ausführlichere Überlegungen zu seiner Planung und Gestaltung enthält der Abschnitt 4.1 der vorliegenden Arbeit.

Die Unterrichtsstunden fanden mittwochs in der 8. Stunde sowie freitags in der 5. und 6. Stunde statt. Für die Doppelstunden an den Freitagen stand ein PC-Pool zur Verfügung, der auch über separate Plätze ohne Computer verfügt. Somit konnten die Schüler in diesen Unterrichtsstunden an den Computern arbeiten, aber auch Unterricht ohne Computernutzung war ohne Beeinträchtigungen möglich.<sup>5</sup>

Stunde	Unterrichtsinhalte, Materialien <sup>6</sup>
1.	<ul style="list-style-type: none"> <li>• Einführung in das Stoffgebiet, Überblick;</li> <li>• Kurzbefragung (anonym): Vektoren und Koordinaten (vgl. S. 300f.);</li> <li>• Wiederholung: Räumliches Koordinatensystem (Folie01-KS.pdf);</li> <li>• Einführung: Funktionsweise der 3D-Computergrafik (Folie02-CG.pdf); zur weiteren Information zu Hause: ABlatt01-Funktionsw-CG.pdf;</li> </ul>
2./3. (PC)	<ul style="list-style-type: none"> <li>• Beschreibung einfacher Körper durch Koordinaten;</li> <li>• Einführung in POV-Ray (Folie03-CG-Beispiel.pdf);</li> <li>• Arbeit am PC: Beschreibung eines Schneemannes durch Koordinaten, Modellierung in POV-Ray (ABlatt02-Kurzanleitung-POVRay.pdf).</li> </ul>
4., 5./6. (PC)	Aufgrund von Kursfahrten waren in dieser Woche nur 6 Schüler anwesend. Die Unterrichtsstunden wurden auf Wunsch dieser Schüler für einige technische und gestalterische Aspekte bei der Arbeit mit POV-Ray (Texturen, Erstellung von Videos) genutzt: ABlatt03-Kurzanleitung-Videos.pdf, ABlatt04-Texturen.pdf, ABlatt05-Stillleben.pdf.
7.	<ul style="list-style-type: none"> <li>• Diskussion von Strategien bei der Modellierung des Schneemannes: Reduzierung der Zahl zu betrachtender Dimensionen, Betrachtung von Schnitten mit Koordinatenebenen (vgl. Abschnitt 4.1.2, Folie04-Schneemann-Schnitt.pdf, Folie05-Dimensionen.pdf);</li> <li>• Herleitung der Kreisgleichung (vgl. 4.1.3, Folie06-Kreisgleichung);</li> </ul>
8./9. (PC)	<ul style="list-style-type: none"> <li>• Herleitung der Kugelgleichung (Folie07-Kugelgleichung.pdf);</li> <li>• Anwendung für exakte Positionierungen der Teile des Schneemannes;</li> <li>• Weiterarbeit an den Grafiken;</li> <li>• Auswertung, Vorstellen der Ergebnisse.</li> </ul>

<sup>5</sup>Doppelstunden sind in der tabellarischen Übersicht durch  $n./m.$ ; Unterrichtsstunden, in denen die Schüler an den Computern arbeiteten, mit (PC) gekennzeichnet.

<sup>6</sup>Die angegebenen Folien, Anleitungen sowie Arbeits- und Aufgabenblätter befinden sich in dem Ordner **Schulversuch-Andreas-Gym**. Es handelt sich um die für die Unterrichtsreihe erarbeiteten, unverändert übernommenen Materialien. Diese entsprechen größtenteils den in den Abschnitten 4.1, 4.2 und 4.6 beschriebenen Aufgaben sowie den in Anhang A enthaltenen Anleitungen, die gegenüber den in dem Schulversuch verwendeten Materialien nur noch leicht überarbeitet wurden.

Nach den skizzierten neun (für die Mehrzahl der Schüler jedoch nur sechs) Unterrichtsstunden musste die Beschäftigung mit der Koordinatenbeschreibung einfacher Körper und der Modellierung in POV-Ray abgebrochen werden, obwohl eine Reihe interessanter Vertiefungs- und Erweiterungsmöglichkeiten bestanden hätte (siehe S. 231f.). Wären diese aufgegriffen worden, so hätte allerdings die verbleibende Zeit für die Behandlung der nach dem Rahmenplan vorgesehenen vektoriellen Raumgeometrie nicht ausgereicht.

Wie bereits in Abschnitt 4.1.6 erwähnt wurde, rief der beschriebene Einstieg in das Stoffgebiet bei vielen Schülern recht großes Interesse hervor, so dass diese Schüler über das für die Anfertigung von Hausaufgaben übliche Maß hinaus in ihrer Freizeit an den Grafiken arbeiteten.<sup>7</sup> Die Aufmerksamkeit einer Reihe von Schülern verlagerte sich schnell von Modellierungsaspekten hin zu der Frage, wie sie ein Video ihres Modells erzeugen können.<sup>8</sup> Einzelne Schüler nahmen in den folgenden Wochen weitere Projekte unter Verwendung von POV-Ray in Angriff, auf die jedoch im Unterricht nicht mehr eingegangen werden konnte. In Pausengesprächen berichteten sie darüber und baten um Rat bei der Lösung einiger dabei auftretender Probleme.

## Vektoren, Geraden und Ebenen

Der auf S. 28ff. wiedergegebene Rahmenplan sieht für das Stoffgebiet Analytische Geometrie in dem Kurs ma-3 hauptsächlich die vektorielle Behandlung von Geraden und Ebenen vor. Der Begriff des Vektors und Elemente der Vektorrechnung sollten bereits in Klasse 11 behandelt worden sein. Wie anhand der Ergebnisse des auf S. 300 beschriebenen Eingangstests deutlich wurde, standen diesbezüglich jedoch unzureichende Voraussetzungen zur Verfügung, so dass auf diese Inhalte eingegangen werden musste. Sowohl für die Einführung von Vektoren als auch bei der Behandlung von Geraden und Ebenen fertigten die Schüler Visualisierungen mithilfe von POV-Ray an (siehe Abschnitt 4.2). Als Schulbücher kamen [27] und ergänzend [26] zum Einsatz; zusätzlich erhielten die Schüler für die Erstellung von Visualisierungen bzw. die visuelle Ergänzung rechnerisch zu lösender Aufgaben Aufgabenblätter mit kurzen Anleitungen.<sup>9</sup> Die folgende Tabelle gibt einen Überblick über den Verlauf der Unterrichtsreihe.

<sup>7</sup>Zu Schülerergebnissen siehe S. 229f. und S. 357ff. sowie die von den Schülern angefertigten Dateien in dem Ordner `Schuelerarbeiten\Schneemaenner-AGym-Berlin`.

<sup>8</sup>Die einfachste Möglichkeit dazu besteht (bei Verwendung der in Abschnitt 3.5 beschriebenen Vorlage) in der zeitabhängigen Darstellung des Blickwinkels (`winkel`) auf die Szene. Einige Schüler generierten auf diese Weise „Rundflüge“ der Kamera um den Schneemann, wobei auch unkorrekte Positionierungen sichtbar wurden, die bei den zuvor betrachteten Ansichten nicht auffielen. Drei von Schülern in diesem Zusammenhang angefertigte Videos enthält der oben (in Fußnote 7) genannte Ordner.

<sup>9</sup>Exemplarisch wurden einige dieser Aufgaben in Abschnitt 4.2 beschrieben. Die vollständigen Aufgabenblätter, die den Schülern zur Verfügung gestellt wurden, sowie zugehörige Lösungsgrafiken befinden sich in dem Ordner `Schulversuch-Andreas-Gym\2-Vektoren+Geraden+Ebenen`. Die in der folgenden Tabelle angegebenen Nummern der Aufgaben folgen der Nummerierung der im Unterricht genutzten Aufgabenblätter (Dateien `Aufgaben-m-n.pdf`). Für die Anfertigung von Visualisierungen kam das Makropaket `anageo.inc` zum Einsatz, das weitgehend dem in Abschnitt 4.2 und in Anhang A auf S. 335f. beschriebenen `anageoR.inc` entspricht (diese Datei wurde lediglich hinsichtlich der Darstellung von Koordinatensystemen gegenüber der älteren Version `anageo.inc` modifiziert).

Stunde	Unterrichtsinhalte, Materialien
10.	<ul style="list-style-type: none"> <li>• Abstände von Punkten, Grundlagen der Vektorrechnung;</li> </ul>
11./12. (PC)	<ul style="list-style-type: none"> <li>• Diskussion und Bestimmung des Vektorbegriffs (siehe S. 200f.);</li> <li>• Veranschaulichung von Vektoren; Orts- und Verbindungsvektoren;</li> <li>• Aufgaben zu Orts- und Verbindungsvektoren, Vektoraddition, vgl. 4.2.1; <a href="#">Anleitung-anageo-inc.pdf</a>, <a href="#">Aufgaben-01-07.pdf</a>;</li> <li>• HA: Aufgabe 7 (Hinführung zu Parameterdarstellungen von Geraden);</li> </ul>
13.	<ul style="list-style-type: none"> <li>• Geradengleichung in Parameterform (ausgehend von Aufgabe 7),<sup>10</sup> <a href="#">Folie-Parameterdarstellung.pdf</a>, <a href="#">Video-Pardarst-Gerade.mpg</a>;</li> <li>• Beschreibung von Geraden durch zwei Punkte;</li> <li>• HA zu Parameter- und 2-Punkte-Gleichungen von Geraden;</li> </ul>
14./15. (PC)	<ul style="list-style-type: none"> <li>• Überprüfung der Zugehörigkeit eines Punktes zu einer Geraden (rechnerisch und visuell, <a href="#">Aufgaben-08-10.pdf</a> – Aufgaben 8, 9);</li> <li>• Parallelität von Geraden: Kriterium, Überprüfung (Aufgabe 10);</li> <li>• Schnittpunkte von Geraden (nur visuell, Aufgabe 10);</li> </ul>
16.	<ul style="list-style-type: none"> <li>• Parametergleichungen von Geraden (Wiederholung, Vertiefung);</li> <li>• Systematisierung: gegenseitige Lage zweier Geraden;</li> <li>• Schnittpunkte von Geraden (rechnerisch);</li> <li>• HA zu Lagebeziehungen von Geraden;</li> </ul>
17.	<ul style="list-style-type: none"> <li>• Lagebeziehungen und Schnittpunkte von Geraden (Wiederholung, Vertiefung, Übungen);</li> </ul>
Herbstferien	
18.	<ul style="list-style-type: none"> <li>• Spurpunkte von Geraden (rechnerisch, visuell: <a href="#">Video-Spurpunkte.mpg</a>);</li> <li>• HA: Bestimmung von Spurpunkten sowie Schnittpunkten von Geraden mit zu Koordinatenebenen parallelen Ebenen (vgl. S. 241 und S. 244);</li> </ul>
19./20. (PC)	<ul style="list-style-type: none"> <li>• Parametergleichungen von Ebenen (<a href="#">Video-Pardarst-Ebene.mpg</a>);</li> <li>• Ebenen durch drei Punkte, Aufstellen von Parametergleichungen;</li> <li>• Visualisierung von Ebenen mithilfe von POV-Ray;</li> <li>• Überprüfung der Zugehörigkeit eines Punktes zu einer Ebene;</li> <li>• Schnittpunkte von Geraden mit Ebenen (visuell);</li> <li>• Aufgaben: <a href="#">Aufgaben-11-16.pdf</a>;</li> <li>• HA: Schnittpunkte von Ebenen mit den Koordinatenachsen, Lage von Ebenen im Raum, Zugehörigkeit von Punkten zu Ebenen;</li> </ul>

<sup>10</sup>Die Aufgabe 7 in den Aufgabenblättern, welche die Schüler bearbeiteten, entspricht der auf S. 237 angegebenen Aufgabe A5. Parametergleichungen wurden nach dem dort beschriebenen Ansatz im Unterricht eingeführt, wobei die Schüler aus Zeitgründen lediglich die Basisversion der Aufgabe bearbeiteten. Verfeinerungen durch eine größere Zahl dargestellter Punkte, `while`-Schleifen und Animationen der Entstehung von Geraden aus Punkten konnten lediglich demonstriert werden.

21./22.	<ul style="list-style-type: none"> <li>• Lagebeziehungen zweier Ebenen;</li> <li>• Lagebeziehungen zwischen Geraden und Ebenen;</li> <li>• Schnittpunkte von Geraden mit Ebenen (rechnerisch);</li> <li>• HA zu Schnittpunkten von Geraden mit Ebenen (rechnerisch und visuell): <a href="#">Aufgaben-17-19.pdf</a>;</li> </ul>
23.	<ul style="list-style-type: none"> <li>• Koordinatengleichungen von Ebenen;</li> <li>• Schnittpunkte von Ebenen mit den Koordinatenachsen;</li> <li>• HA: Aufstellen von Koordinatengleichungen von Ebenen, die durch Parametergleichungen gegeben sind;</li> </ul>
24./25.	<ul style="list-style-type: none"> <li>• Umformung von Ebenengleichungen: Koordinatengleichung <math>\rightarrow</math> Parametergleichung;</li> <li>• Wiederholung, Aufgaben zur Vorbereitung auf die Klausur;</li> </ul>
26./27.	<ul style="list-style-type: none"> <li>• Klausur (90 Minuten), siehe <a href="#">Klausur.pdf</a>, <a href="#">Klausurergebnisse.pdf</a>;</li> </ul>
28.	<ul style="list-style-type: none"> <li>• Rückgabe und Auswertung der Klausur.</li> </ul>

Wie aus der Übersicht hervorgeht, mussten in der zur Verfügung stehenden Zeit aufgrund der Rahmenplanvorgaben recht viele Typen von Standardaufgaben behandelt werden;<sup>11</sup> dadurch stand wenig Zeit für Vertiefungen und über den Rahmenplan hinausgehende Inhalte zur Verfügung. So erfolgte die Einführung von Parameterdarstellungen zwar unter Berücksichtigung der in Abschnitt 3.3.2 dargestellten Überlegungen; aufgegriffen und vertieft werden konnten funktionale Aspekte im Zusammenhang mit Parameterdarstellungen im folgenden Unterricht jedoch nicht mehr; auch für die Beschreibung von Kurven durch Parameterdarstellungen war keine Zeit vorhanden. Es ist somit zu bezweifeln, dass die Schüler ein fundiertes Verständnis hinsichtlich des Punktmengencharakters von Geraden und Ebenen sowie funktionaler Zusammenhänge zwischen Positionen von Punkten im Raum und Parametern erlangen konnten.

Visualisierungen erstellten die Schüler im Unterricht aus Zeitgründen nur exemplarisch für einige Aufgaben. Viele Schüler nutzten jedoch die Möglichkeit, Aufgaben der analytischen Geometrie mithilfe von POV-Ray grafisch darzustellen, auch für ihre Hausaufgaben, obwohl dies nicht verlangt war. Als Grund gaben sie an, dass sie sich so unter den Aufgaben mehr vorstellen und vor allem ihre rechnerischen Ergebnisse kontrollieren können. Zusätzlich zu Computervisualisierungen fertigten die Schüler insbesondere im Zusammenhang mit der Behandlung der Spurpunkte von Geraden sowie der Schnittpunkte von Ebenen mit den Koordinatenachsen auch Handskizzen an.

<sup>11</sup>Größtenteils wurden für diese Aufgaben Geraden und Ebenen gewählt, die zu relativ leicht lösbaren Gleichungssystemen führen, da die Mehrzahl der Schüler erhebliche Schwierigkeiten bei der Lösung linearer Gleichungssysteme hatte. Wie bereits erwähnt wurde, konnte für die Behandlung der LGS am Ende von Klasse 12 nicht die vorgesehene Stundenzahl genutzt werden; eine sehr ausführliche Wiederholung und Weiterführung dieses Themengebietes hätte unweigerlich zu Abstrichen an den vorgesehenen Inhalten der analytischen Geometrie geführt.



In der *Klausur* (deren Termin zentral festgelegt war und die deshalb nicht am Ende des gesamten Stoffgebietes geschrieben werden konnte) nahmen die nach dem Rahmenplan vorgesehenen Inhalte den größten Raum ein; auf die im ersten Teil der Unterrichtsreihe behandelte Koordinatengeometrie bezog sich nur eine Aufgabe, für die 17,5% der Punkte vergeben wurden. Insgesamt bestand die Klausur aus 4 Aufgaben zu folgenden Themengebieten:<sup>12</sup>

1. Lagebeziehungen von Geraden (30% / 72%);<sup>13</sup>
2. Ebenengleichungen, Lage einer Ebene im Raum, Schnittpunkte mit den Koordinatenachsen, Schnittpunkt einer Gerade und einer Ebene (40% / 71%);
3. Aufgabe zur Vektorrechnung: Bestimmung der fehlenden Eckpunkte eines Parallelogramms, von dem zwei Eckpunkte und der Schnittpunkt der Diagonalen gegeben sind (12,5% / 36%);
4. Anordnungen von Kugeln im Raum (17,5% / 53%).<sup>14</sup>

### Skalarprodukt und Normalenvektoren unter Berücksichtigung von Anwendungen in der Computergrafik

Für den letzten Teil der Unterrichtsreihe standen noch maximal 6 Unterrichtsstunden zur Verfügung. Insofern war es nicht möglich, alle in Abschnitt 4.6 beschriebenen Aspekte der Einbeziehung mathematischer Grundlagen der Computergrafik umzusetzen; u. a. musste darauf verzichtet werden, lokale Beleuchtungsmodelle zu behandeln. Zumindest sollten die prinzipielle Funktionsweise des Raytracing-Verfahrens und im Zusammenhang damit das Reflexionsgesetz im Raum den Einstieg in diesen Teil der Unterrichtsreihe bilden (vgl. 4.6.1). Weiterhin wurden bei der Erarbeitung des Skalarproduktes und von Normalenvektoren (wie in 4.6.2 und 4.6.3 beschrieben) von den Schülern Visualisierungen angefertigt und auf deren Grundlage Hypothesen aufgestellt. Schließlich modellierten die Schüler ein Oktaeder mithilfe von Dreiecken und „glätteten“ seine Kanten durch die Berechnung und Mittelung der Normaleneinheitsvektoren der Ebenen, denen die Seitenflächen des Oktaeders angehören (siehe 4.6.6). Dieses Beispiel kombiniert das Aufstellen von Ebenengleichungen und Berechnungen von Normalenvektoren, welche die Schüler nach dem Rahmenplan ohnehin durchzuführen hatten, mit einer interessanten Anwendung in der Computergrafik, die zu einem recht verblüffenden Ergebnis führt. Da der zusätzliche Zeitaufwand hierbei gegenüber der Lösung von Standardaufgaben relativ gering ist, erschien die Einbeziehung einer solchen Aufgabe in den Unterricht trotz der Zeitknappheit sinnvoll. Im Folgenden wird ein Überblick über den Verlauf des letzten Teils der Unterrichtsreihe gegeben.

---

<sup>12</sup>Die vollständigen Klausuraufgaben sowie eine (anonymisierte) Zusammenstellung der von den Schülern erreichten Ergebnisse enthalten die Dateien `Klausur.pdf` und `Klausurergebnisse.pdf` in dem Ordner `Schulversuch-Andreas-Gym\2-Vektoren+Geraden+Ebenen` (siehe Anhang D, S. 365).

<sup>13</sup>Die Werte in Klammern ( $n\% / m\%$ ) geben den Anteil  $n$  der für die jeweilige Aufgabe vergebenen Punkte an der insgesamt in der Klausur erreichbaren Punktzahl sowie den durchschnittlichen Prozentsatz  $m$  der von den Schülern erreichten Punkte für die Aufgabe an.

<sup>14</sup>Die vierte Aufgabe wurde von 4 Schülern offenbar aus Zeitgründen nicht in Angriff genommen. Diejenigen Schüler, welche die Aufgabe bearbeiteten, erreichten durchschnittlich 66% der Punkte.

Stunde	Unterrichtsinhalte, Materialien
29./30. (PC)	<ul style="list-style-type: none"> <li>• Überblick: Funktionsweise des Raytracing-Verfahrens;</li> <li>• Wiederholung: Reflexionsgesetz;</li> <li>• Einführung: Skalarprodukt zweier Vektoren;</li> <li>• Orthogonalität und Winkel zwischen Vektoren (visuell): Aufgabe 20 des Aufgabenblattes <b>Aufgaben-20-21.pdf</b>;<sup>15</sup></li> <li>• Herausarbeiten des Zusammenhangs zwischen dem Skalarprodukt und dem Winkel zweier Vektoren;</li> <li>• HA: Aufgabe 21 (Hinführung zu Normalenvektoren);</li> </ul>
31.	<ul style="list-style-type: none"> <li>• Auswertung der HA; Einführung: Normalenvektoren;</li> <li>• Koordinatenform der Ebenengleichung und Normalenvektoren;</li> <li>• Skalarprodukte von Normalen- und Richtungsvektoren;</li> <li>• Normieren von Vektoren;</li> <li>• HA: Normaleneinheitsvektoren der Seitenflächen eines Oktaeders berechnen (Aufgabe 22, Teile c und d);<sup>16</sup></li> </ul>
32./33. (PC)	<ul style="list-style-type: none"> <li>• Vergleich der HA;</li> <li>• Überblick: Glättung von Kanten (Beispiel: Fisch, siehe S. 136ff.);</li> <li>• Darstellung eines Oktaeders in POV-Ray durch die Schüler;</li> <li>• Bestimmung gemittelter Normaleneinheitsvektoren, Darstellung eines Oktaeders mit geglätteten Kanten (Aufgabe 22, Teile e, f und g);</li> <li>• <b>Aufgabe-22.pdf</b>, <b>Folie-Oktaeder.pdf</b>, <b>Oktaeder-Glatt-Video.mpg</b>;</li> </ul>
34.	<ul style="list-style-type: none"> <li>• Wiederholung, Zusammenfassung;</li> <li>• Abschlussbefragung (siehe S. 309ff.).</li> </ul>

Nach diesen Unterrichtsstunden musste das Stoffgebiet Analytische Geometrie beendet werden, um für die Behandlung des 2. Stoffgebietes innerhalb des Kurses ma-3 (Gebrochenrationale Funktionen) noch ausreichend Zeit zu behalten. Das recht geringe Maß an zur Verfügung stehender Zeit zwang zu Kompromissen bei der Auswahl des zu behandelnden Stoffes. So erfolgte die Einführung des Skalarproduktes recht zügig unter entsprechend deutlichen Vorgaben des Lehrers. Den qualitativen Zusammenhang zwischen dem Skalarprodukt und dem Winkel zweier Vektoren erkannten die Schüler nach der Bearbeitung von Aufgabe 20 zwar selbstständig, die Gleichung  $\langle \vec{a}, \vec{b} \rangle = |\vec{a}| \cdot |\vec{b}| \cdot \cos \angle(\vec{a}, \vec{b})$  wurde jedoch dann durch den Lehrer vorgegeben, obwohl bei mehr zur Verfügung stehender Zeit eine Erarbeitung durch die Schüler sicherlich möglich gewesen wäre. Auch auf einen Beweis wurde aus Zeitgründen verzichtet. Bei der Lösung von Aufgabe 22 arbeiteten die Schüler in Gruppen, wobei einige Schüler Berechnungen und andere die Eingaben in POV-Ray vornahmen. Dadurch konnte die Aufgabe im Unterricht zu Ende geführt werden, wobei das Ergebnis großes Erstaunen der Schüler hervorrief.

<sup>15</sup>Aufgabe 20 entspricht Aufgabe A auf S. 278; Aufgabe 21 ist identisch mit Aufgabe B auf S. 280.

<sup>16</sup>Die Aufgabe 22 ist identisch mit der Aufgabe E auf S. 287.

### 5.1.3 Abschlussbefragung der Schüler zur Einbeziehung von Elementen der 3D-Computergrafik

In der letzten Unterrichtsstunde zur analytischen Geometrie (am 3. Dezember 2003) füllten die 14 in dieser Stunde anwesenden Schüler einen (anonymen) Fragebogen aus, mit dem ihre Meinungen zur Gestaltung des Stoffgebietes unter Einbeziehung von Elementen der Computergrafik in Erfahrung gebracht werden sollten. Durch Prozentangaben von 0% („Aussage trifft überhaupt nicht zu“) bis 100% („Aussage trifft in vollem Umfang zu“) wurde das Maß an Zustimmung der Schüler zu den im Folgenden aufgeführten Aussagen erfasst.<sup>17</sup>

#### 1. Allgemeine Fragen

Zustimmung:	0%	25%	50%	75%	100%	Ich weiß nicht
Die Ergänzung des Stoffgebietes Analytische Geometrie um Elemente der 3D-Computergrafik ist sinnvoll.	–	–	–	8	5	1
Es wäre schön gewesen, wenn mehr Zeit für die Beschäftigung mit Themen der Computergrafik zur Verfügung gestanden hätte.	–	1	1	9	3	–
Die Einbeziehung der Computergrafik lenkt vom Wesentlichen – dem Lernen dessen, was man aus der analytischen Geometrie braucht – ab.	5	4	3	1	–	1

#### 2. Fragen zum Einstieg in das Stoffgebiet

Zustimmung:	0%	25%	50%	75%	100%	Ich weiß nicht
Koordinatenbeschreibungen von Körpern und Modellierung in POV-Ray (Schneemann) sind ein guter Einstieg in das Stoffgebiet Analytische Geometrie.	–	2	2	1	9	–
Der Zusammenhang zwischen der anfänglichen Arbeit mit POV-Ray und den anderen Inhalten des Stoffgebietes ist vorhanden und sichtbar.	–	1	7	3	1	2
Die Zeit, die am Anfang für die POV-Ray-Einführung, die Koordinatenbeschreibung von Körpern und den Schneemann verwendet wurde, hätte an anderen Stellen des Stoffgebietes sinnvoller genutzt werden können.	6	2	2	2	2	–

<sup>17</sup>Kopien aller von den Schülern ausgefüllten Fragebögen enthält die Datei **Abschlussbefragung-Antworten-Scans.pdf** in dem Ordner **Schulversuch-Andreas-Gym** (siehe S. 365).

### 3. Fragen zur Visualisierung von Punkten, Vektoren, Geraden und Ebenen

Zustimmung:	0%	25%	50%	75%	100%	Ich weiß nicht
Die Visualisierung von Objekten der analytischen Geometrie (Vektoren, Geraden, Ebenen) in POV-Ray hilft, diese Objekte besser zu verstehen.	–	1	–	8	5	–
Durch die Darstellung in POV-Ray ist mir klarer geworden, was Vektoren eigentlich sind.	1	–	1	4	7	1
Durch die Erarbeitung mithilfe von POV-Ray habe ich besser verstanden, was Parametergleichungen von Geraden bedeuten.	1	3	4	3	2	1
Der geometrische Zusammenhang zwischen Geraden und Ebenen sowie den zugehörigen Richtungsvektoren ist durch POV-Ray deutlicher geworden.	–	1	4	4	4	1
Die Einbeziehung der 3D-Computergrafik hat Zeit gekostet. Hätte diese Zeit für eine größere Zahl von Übungen zur Verfügung gestanden, hätte ich Aufgaben der analytischen Geometrie besser lösen können.	4	3	–	2	3	2
Die Veranschaulichung von Aufgaben der analytischen Geometrie mittels Computer trägt besser zum Verständnis bei als eine größere Zahl von rechnerisch zu lösenden Aufgaben.	1	–	5	5	2	1
Die Darstellung am Computer bringt mehr anschauliches Verständnis als Skizzen an der Tafel und die Verwendung herkömmlicher Modelle (Koordinatensystem, Lineale, Stifte).	2	–	1	3	8	–

### 4. Fragen zur Funktionsweise der 3D-Computergrafik im Zusammenhang mit Skalarprodukt und Normalenvektoren

Zustimmung:	0%	25%	50%	75%	100%	Ich weiß nicht
Beispiele aus der 3D-Computergrafik haben dazu beigetragen, dass ich Anwendungen von Begriffen der analytischen Geometrie (wie „Normalenvektor“) gesehen habe.	–	1	2	1	4	6
Anwendungen der analytischen Geometrie in der Computergrafik tragen dazu bei, die analytische Geometrie als „sinnvolles“ Thema anzusehen.	–	–	4	6	1	3

Zusätzlich wurden zwei Fragen gestellt, die von den Schülern verbal zu beantworten waren:

- 5. Was hätten Sie sich anders gewünscht? Evtl. mit Vorschlägen: Wie?**
- 6. Hat die Einbeziehung der Computergrafik in den Mathematikunterricht dazu beigetragen, Ihr Bild von der Mathematik zu verändern? Wenn ja, in welcher Hinsicht?**

Auf diese beiden Fragen gaben die Schüler folgende Antworten:

5. Ich hätte mir für das Darstellen am Computer mehr Zeit gewünscht. Wir haben leider nur einen kleinen Eindruck bekommen in die 3D-Computergrafik, so haben wir zum Beispiel aufgrund einer Klausur kein Video machen können.
6. Ja es hat meine Sicht verändert, denn es war eine gute Abwechslung zur Theorie und ich habe dadurch mehr verstanden.
5. Ich hätte mir insgesamt mehr Zeit gewünscht, vor allem für die Videos und Schneemänner.
6. Ich würde dem zustimmen. Meine Vorstellungen von einer trockenen, fast nur theoretischen Mathematik haben sich durch die praktischen Anwendungen von POV-Ray ein wenig geändert.
5. Die Theorie hätte trotzdem mehr im Vordergrund stehen sollen. Im Endeffekt konnte man Thema dennoch verstehen und die Anschaulichkeit war gut gelöst.
6. –
5. – 6. –
5. Computerbearbeitete Aufgaben hätten auch bearbeitet werden sollen (wie? schwer zu sagen...)
6. ja! zum ersten Mal fassbar, vorstellbar & greifbar!
5. mehr Zeit
6. naja, Mathe bleibt Mathe
5. –
6. Mein Bild von Mathematik hat sich nicht verändert, aber gerade das Thema der anal. Geometrie lässt sich mit Hilfe von PCs und POV-Ray besser darstellen und veranschaulichen.
5. Alles supi. Machen Sie weiter so.
6. Teilweise schon, aber manche Themen lassen sich nicht so gut visualisieren wie die anal. Geometrie.

5. Ich denke, dass ich leider zu selten am Unterricht teilnehmen konnte und ich deshalb keine Beurteilung an der Unterrichtsgestaltung geben kann.
6. –
5. Es wäre sinnvoller gewesen, die praktische Arbeit an die Theoriearbeit anzuschließen. Theoriearbeit sollte als Basis dienen, um dann veranschaulicht und intensiviert zu werden.
6. nicht wesentlich
5. mehr Zeit mit dem CPU
6. eher nicht
5. weniger Ausfallstunden am Freitag!
6. Ja, da der Unterricht anschaulicher geworden ist. Die Individualität der Schüler konnte gut berücksichtigt werden, das hat mir sehr gefallen. Ich werde mich bestimmt gerne daran zurückerinnern.
5. Noch ein Beispiel mehr als Schneemann und nicht dann nur noch Geraden und Ebenen...
6. Ist auf jeden Fall mal was anderes  $\leftrightarrow$  „sinnvolle“ Erscheinung wird deutlich  $\rightarrow$  erschien sonst nur als sinnloser Zahlenkomplex, wobei eindeutiger Nutzen noch nicht 100%ig klar geworden ist.  $\rightarrow$  Anwendungsgebiet wird deutlich  $\rightarrow$  Sinn nicht!
5. –
6. Durch die Computergrafik war vieles einfacher und besser zu verstehen und so hat Mathematik auch mal Spaß gemacht.

#### 5.1.4 Zusammenfassung und einige Schlussfolgerungen

Die von den Schülern gegebenen Antworten bringen m. E. zum Ausdruck, welche mit der Einbeziehung von Elementen der Computergrafik in das Stoffgebiet Analytische Geometrie verbundenen Möglichkeiten in dem beschriebenen Unterrichtsversuch umgesetzt werden konnten. Gleichzeitig verdeutlichen die Antworten Defizite des Aufbaus der Unterrichtsreihe. Trotz eines recht differenzierten Meinungsbildes hinsichtlich vieler Fragen sind folgende Tendenzen klar zu erkennen:

- Die Einbeziehung von Elementen der Computergrafik stieß bei der Mehrheit der Schüler auf Interesse und bewirkte eine höhere Motivation für den Unterricht.
- Die Schüler konnten zumindest in Ansätzen erkennen, dass Verfahren der 3D-Computergrafik interessante Anwendungen von Unterrichtsinhalten des Stoffgebietes Analytische Geometrie darstellen.
- Der Anschaulichkeit des Unterrichts kam die Erstellung von Computervisualisierungen in erheblichem Maße zugute.

- Es traten inhaltliche Brüche innerhalb der Unterrichtsreihe auf. Insbesondere der Zusammenhang zwischen dem koordinatengeometrischen Einstieg in das Stoffgebiet und der folgenden vektoriellen Geometrie von Geraden und Ebenen konnte somit nicht hinreichend deutlich werden (vgl. insbesondere die Antworten auf die zweite Frage unter „2. Fragen zum Einstieg in das Stoffgebiet“ auf S. 309).
- Die beschriebene Einführung von Parameterdarstellungen führte offensichtlich bei vielen Schülern nicht zu einem tieferen Verständnis dieser Art der Beschreibung von Punktmengen (vgl. auch die entsprechenden Anmerkungen auf S. 306).
- Die Einbeziehung von Elementen der Computergrafik führte dazu, dass für das rechnerische Lösen der nach dem Rahmenplan geforderten (und dementsprechend auch die Klausur dominierenden) Standardaufgaben weniger Zeit aufgewendet wurde. Insbesondere die Antworten auf die dritte Frage unter 2. auf S. 309 sowie auf die fünfte Frage unter 3. auf S. 310 zeigen, dass einige Schüler dies als problematisch ansahen, wenngleich die überwiegende Mehrheit angab, dass Visualisierungen von Vektoren, Geraden und Ebenen zu mehr Verständnis dieser Objekte führten (vgl. die ersten beiden Fragen unter 3. auf S. 310).
- In der kurzen Zeit, die für die Behandlung des Skalarproduktes und von Normaleneinheitsvektoren verblieb, konnten die mit der Einbeziehung von Elementen der 3D-Computergrafik verfolgten Ziele (vgl. 4.6) nur ansatzweise erreicht werden. Dies wurde bereits im Verlauf der letzten Stunden der Unterrichtsreihe deutlich und geht auch aus den zum großen Teil recht „unentschiedenen“ Antworten auf die Fragen zu 4. (S. 310) hervor.

Zusammenfassend ist m. E. festzustellen, dass Möglichkeiten der Einbeziehung von Elementen der Computergrafik in das Stoffgebiet – insbesondere hinsichtlich der Motivierung der Schüler sowie einer anschaulicheren Unterrichtsgestaltung – in der beschriebenen Unterrichtsreihe teilweise umgesetzt werden konnten. Allerdings erlangten die Schüler hinsichtlich wichtiger Inhalte, z. B. der Beschreibung einfacher Körper durch Gleichungen sowie funktionaler Aspekte von Parameterdarstellungen, kein hinreichend tiefes Verständnis und bildeten nicht die mit der Behandlung dieser Inhalte angestrebten Fähigkeiten aus. Dies ist aufgrund der nur ansatzweisen, unter Zeitdruck erfolgten Behandlung dieser Gegenstände nicht erstaunlich. Größere Erfolge hätten bei einem höheren Maß an zur Verfügung stehender Zeit sicherlich erzielt werden können. Allerdings geht aus dem in Abschnitt 5.1.2 dargestellten Verlauf der Unterrichtsreihe bereits hervor, dass der größte Teil der Zeit für die Behandlung von Standardaufgaben der vektoriellen Raumgeometrie aufgewendet werden musste.

Eine Reduzierung der von den Rahmenplänen geforderten Zahl an „Aufgabeninseln“ (vgl. Abschnitt 1.4.2) und eine Vertiefung von Kerninhalten könnte, wie bereits mehrfach ausgeführt wurde, zu einem tieferen inhaltlichen Verständnis und einer größeren Transferbreite dieser Inhalte führen. Nach den hier beschriebenen Erfahrungen wäre dadurch auch eine wichtige Voraussetzung dafür gegeben, die mit der Einbeziehung von Elementen der Computergrafik verbundenen Potenzen in höherem Maße zu nutzen.

## 5.2 Unterrichtsversuch an der Internationalen Gesamtschule Heidelberg

Im Frühsommer 2005 führte J. KÖBERLE im Rahmen seiner schriftlichen Prüfungsarbeit [138] zur zweiten Staatsprüfung für das Lehramt an Gymnasien eine Unterrichtsreihe zum Einstieg in die analytische Geometrie unter Einbeziehung von Elementen der 3D-Computergrafik durch. Die 14 Unterrichtsstunden umfassende Reihe fand in einer 12. Klasse<sup>18</sup> der Internationalen Gesamtschule Heidelberg (IGH) nach den 2005 für die gymnasiale Oberstufe noch gültigen Rahmenplänen statt.<sup>19</sup> Im Folgenden wird ein kurzer Überblick über die von KÖBERLE erteilten Unterrichtsstunden gegeben; ausführliche Beschreibungen finden sich in der bereits erwähnten Arbeit [138].<sup>20</sup>

Stunde	Unterrichtsinhalte
1.	<ul style="list-style-type: none"> <li>• Einführung in das Stoffgebiet;</li> <li>• Informationen zur Arbeitsweise der 3D-Computergrafik;</li> <li>• Arbeit der Schüler mit POV-Ray (Einstieg in die Nutzung der Software);</li> </ul>
2.	<ul style="list-style-type: none"> <li>• Modellierung eines Schneemanns mithilfe von POV-Ray;</li> </ul>
3.	<ul style="list-style-type: none"> <li>• Vertiefung: Punkte im räumlichen Koordinatensystem;</li> <li>• Vereinfachung räumlicher Positionierungsprobleme;</li> <li>• Weiterführung der Arbeit mit POV-Ray;</li> </ul>
4.	<ul style="list-style-type: none"> <li>• Verschiebungen;</li> <li>• Einführung des Vektorbegriffs, Addition von Vektoren;</li> </ul>
5./6.	<ul style="list-style-type: none"> <li>• Multiplikation von Vektoren mit Skalaren;</li> <li>• Verbindungsvektoren zweier Punkte;</li> <li>• Visualisierung von Vektoren durch Pfeile (vgl. Abschnitt 4.2.1);</li> <li>• Vergleich der Ergebnisse bei der Modellierung des Schneemanns;<sup>21</sup></li> </ul>

<sup>18</sup>In Baden-Württemberg wird der Mathematikunterricht in der gymnasialen Oberstufe nicht mehr differenziert in Grund- und Leistungskursen durchgeführt.

<sup>19</sup>Siehe die Datei `BadenWürttemberg-Alt-12-13.pdf` in dem Ordner `Rahmenplaene` (siehe S. 365).

<sup>20</sup>Diese Arbeit und ein Anhang mit verwendeten Unterrichtsmaterialien und Leistungsüberprüfungen befinden sich in dem Ordner `Pruefungsarbeit-J-Koeberle\Pädagogische Arbeit`. Herr KÖBERLE erklärte sich ausdrücklich mit der Veröffentlichung seiner Arbeit und der Materialien im Rahmen der vorliegenden Arbeit einverstanden. Die Arbeitsblätter und Vorlagen, die er den Schülern zur Verfügung stellte, sowie die gestellten Aufgaben entsprachen zum großen Teil denjenigen, die bereits in Abschnitt 5.1 beschrieben wurden, bzw. entstanden aus diesen durch kleinere Variationen; sie werden daher in der folgenden Tabelle nicht mehr explizit aufgeführt.

<sup>21</sup>Einige Ergebnisse sind auf S. 359f. abgedruckt; die von den Schülern angefertigten POV-Ray-Dateien befinden sich in dem Ordner `Pruefungsarbeit-J-Koeberle\Schneemänner`. Auf die außergewöhnlich ausgereifte Arbeit eines Schülers der Klasse, der u. a. Verschiebungen, Drehungen und zentrische Streckungen für die Anordnung mehrerer Exemplare eines Schneemanns nutzte (obwohl diese



7.	<ul style="list-style-type: none"> <li>• Einführung: Parameterdarstellungen von Geraden (vgl. A5 auf S. 237);</li> <li>• Aufgaben zu Parameterdarstellungen (rechnerisch und visuell);</li> </ul>
8./9.	<ul style="list-style-type: none"> <li>• Beschreibung von Geraden durch Parametergleichungen;</li> <li>• Zugehörigkeit von Punkten zu Geraden;</li> <li>• Lagebeziehungen von Geraden (rechnerisch und visuell, vgl. S. 238ff.);</li> <li>• Lösen linearer Gleichungssysteme (LGS) mithilfe eines GTR;<sup>22</sup></li> </ul>
10.	<ul style="list-style-type: none"> <li>• Lagebeziehungen von Geraden, Aufstellen und Lösen von LGS;</li> <li>• Einführung / Veranschaulichung: Parameterdarstellungen von Ebenen;</li> </ul>
11./12.	<ul style="list-style-type: none"> <li>• Festigung: Parameterdarstellungen von Ebenen;</li> <li>• Aufgaben zur Beschreibung und Visualisierung von Ebenen;</li> <li>• Zugehörigkeit von Punkten zu Ebenen (rechnerisch und visuell);</li> <li>• Schnittpunkte von Ebenen mit den Koordinatenachsen;</li> <li>• Lagebeziehungen von Geraden und Ebenen (rechnerisch und visuell);</li> </ul>
13.	<ul style="list-style-type: none"> <li>• Wiederholung, Übungen zur Vorbereitung auf die Lernzielkontrolle (Aufgaben zu allen Themenbereichen der Unterrichtsreihe; für die Lösungen wurde teilweise POV-Ray genutzt);</li> </ul>
14.	<ul style="list-style-type: none"> <li>• Lernzielkontrolle.<sup>23</sup></li> </ul>

Der Verlauf und die Ergebnisse der Unterrichtsreihe wurden von KÖBERLE überwiegend positiv eingeschätzt (vgl. [138], S. 32f.). Die Mehrzahl der Schüler begrüßte den Einstieg in die Unterrichtsreihe über die Beschreibung von Objekten durch Koordinaten sowie die Möglichkeit, auch die Inhalte des folgenden Unterrichts zu veranschaulichen. Einige Schüler fanden besondere Freude an der Arbeit mit POV-Ray und nutzten die Software weit über das geforderte Maß hinaus auch in ihrer Freizeit.<sup>24</sup> KÖBERLE hob weiterhin hervor, dass auch Schüler, die sich sonst kaum am Mathematikunterricht beteiligten, recht aktiv mitarbeiteten und bei der Lösung von Aufgaben an den Computern angeregt mit ihren Partnern über die Vorgehensweisen diskutierten.

Abbildungen kein Unterrichtsgegenstand waren), wurde bereits kurz auf S. 258 eingegangen. Zwei Schüler beschäftigten sich offenbar selbstständig mit der Beschreibung von Farben durch Zahlentripel (auch diese Thematik kam im Unterricht nicht zur Sprache), beschränkten sich also nicht auf die Nutzung der vorgegebenen Texturen (siehe die Dateien `Schüler_7.2.pov` und `Schüler_9.2.pov`).

<sup>22</sup>Es wurde der grafische Taschenrechner (GTR) TI-83 Plus verwendet, unter dessen Verwendung sich lineare Gleichungssysteme lösen lassen. Die Schüler waren mit der Bedienung dieses Gerätes bereits aus dem vorangegangenen Unterricht vertraut.

<sup>23</sup>Drei der sechs in der Lernzielkontrolle gestellten Aufgaben lösten die Schüler an Computern unter Verwendung von POV-Ray. Die Aufgaben der Lernerfolgskontrolle und einige der von Schülern angefertigten Lösungen befinden sich im Anhang der Arbeit [138] von KÖBERLE (S. 42ff.), eine Auswertung der Ergebnisse auf S. 28ff. seiner Arbeit.

<sup>24</sup>In einer Zusatzveranstaltung außerhalb der regulären Unterrichtszeit führte KÖBERLE interessierte Schüler in die Erstellung von Videos mithilfe von POV-Ray ein.

Am Ende der Unterrichtsreihe füllten die Schüler den folgenden Fragebogen aus, der dem in Abschnitt 5.1.3 dargestellten recht ähnlich ist; die Fragen unterscheiden sich hauptsächlich aufgrund abweichender Unterrichtsinhalte (vgl. [138], S. 30f.).

### Allgemeines

Zustimmung:	0%	25%	50%	75%	100%	Ich weiß nicht
Die Arbeit am Computer während der Unterrichtseinheit hat mir Spaß gemacht.	2	1	3	4	8	–
Die Verwendung eines 3D-Computergrafik-Programms in der analytischen Geometrie ist sinnvoll.	1	–	3	1	13	–
Die Verwendung der Computergrafik lenkt vom Lernen dessen, was man aus der analytischen Geometrie braucht, ab.	8	3	2	4	–	1
Die Anleitung zur Verwendung des Programms POV-Ray war verständlich.	–	1	1	1	15	–

### Einstieg in das Stoffgebiet

Das Beschreiben von Objekten durch Koordinaten und das Erstellen eines Schneemannes sind ein guter Einstieg in das Stoffgebiet Analytische Geometrie.	–	–	3	7	8	–
Der Zusammenhang zwischen der anfänglichen Arbeit mit POV-Ray und den anderen Inhalten des Stoffgebietes ist vorhanden und sichtbar.	1	3	3	4	7	–
Die Zeit, die am Anfang für die POV-Ray-Einführung und den Schneemann verwendet wurde, hätte an anderen Stellen sinnvoller genutzt werden können.	7	3	3	2	3	–

### Punkte, Vektoren, Geraden

Die Visualisierung von Objekten der analytischen Geometrie (Vektoren, Geraden, Ebenen) in POV-Ray hilft, diese Objekte besser zu verstehen.	–	1	2	2	13	–
Durch die Darstellung in POV-Ray ist mir klarer geworden, was Vektoren eigentlich sind.	1	–	2	4	11	–
Durch die Erarbeitung mithilfe von POV-Ray habe ich besser verstanden, was Parameterdarstellungen von Geraden bedeuten.	3	2	2	4	7	–
Der geometrische Zusammenhang zwischen Geraden und Ebenen sowie den zugehörigen Richtungsvektoren ist durch POV-Ray deutlich geworden.	1	2	1	5	9	–

Zustimmung:	0%	25%	50%	75%	100%	Ich weiß nicht
Mit mehr Übungen und weniger Beschäftigung mit der 3D-Computergrafik hätte ich Aufgaben der analytischen Geometrie besser lösen können.	3	3	3	4	4	1
Die Veranschaulichung von Aufgaben der analytischen Geometrie mithilfe des Computers trägt besser zum Verständnis bei als eine größere Zahl von rechnerisch zu lösenden Aufgaben.	–	1	4	4	8	1
Die Darstellung am Computer bringt mehr anschauliches Verständnis als Skizzen an der Tafel.	–	–	3	3	12	–

Obwohl die Einschätzungen der Schüler hinsichtlich der Auswirkungen, welche die Anfertigung von Computervisualisierungen für ihr Verständnis zentraler Inhalte des Stoffgebietes hatte, insgesamt noch etwas positiver ausfielen als bei der in Abschnitt 5.1.3 dargestellten Erhebung, lassen sich sehr ähnliche grundsätzliche Tendenzen feststellen. Die in Abschnitt 5.1.4 gegebenen Zusammenfassungen treffen, soweit sie sich auf Inhalte beziehen, die auch in der kürzeren Unterrichtsreihe von KÖBERLE behandelt wurden, weitgehend auch für den hier betrachteten Unterrichtsversuch zu. Jedoch fallen bei Betrachtung der Antworten zwei Besonderheiten besonders deutlich auf:

- Die Antworten auf die ersten beiden Fragen unter „Allgemeines“ (S. 316) lassen den Schluss nahe liegend erscheinen, dass auch einige Schüler, denen die Arbeit am Computer keine besondere *Freude* bereitete, der Ansicht waren, dass die Verwendung von POV-Ray im Unterricht der analytischen Geometrie *sinnvoll* ist.<sup>25</sup>
- Das Meinungsbild hinsichtlich der ersten, fünften und sechsten Frage unter „Punkte, Vektoren, Geraden“ lässt vermuten, dass einige Schüler zumindest unterschwellig einen deutlichen Unterschied zwischen dem *Verständnis* mathematischer Inhalte und der *Fähigkeit, Aufgaben zu lösen*, sahen. Es liegt die Schlussfolgerung nahe, dass diese Schüler zu der Einschätzung gelangten, die Einbeziehung von Computervisualisierungen habe zwar ihr Verständnis zentraler Gegenstände des Unterrichts gefördert, jedoch weniger zu ihrer Fähigkeit beigetragen, Aufgaben der analytischen Geometrie zu lösen.<sup>26</sup>

<sup>25</sup>Interessante Erfahrungen legte KÖBERLE in Bezug auf einzelne Schüler dar, die über gering ausgeprägte Computerkenntnisse verfügten und auch wenig Begeisterung für die Nutzung des Computers im Mathematikunterricht zeigten. Zumindest zum Teil konnten diese Schüler im Verlauf der Unterrichtsreihe ihre Fähigkeiten im Umgang mit dem Computer deutlich ausbauen und entwickelten etwas Interesse für die Arbeit mit POV-Ray. Auch wenn es sich hierbei nicht um ein zentrales Ziel des Mathematikunterrichts handelt, kann diese Tatsache zumindest als positiver Nebeneffekt angesehen werden. Die relativ offene Aufgabenstellung, einen Schneemann zu modellieren, wirkte sich in dieser Hinsicht günstig aus, da sie Lösungen sehr unterschiedlichen Schwierigkeitsgrades zulässt und somit auch Schülern, die relativ viel Zeit für die Einarbeitung benötigen, Erfolgserlebnisse ermöglicht.

<sup>26</sup>Die Differenzierung zwischen Verständnis und der Fähigkeit, Aufgaben zu lösen, könnte zumindest teilweise aus Vorerfahrungen der Schüler mit der Lösung von Routineaufgaben resultieren (vgl. 1.4.2).

### 5.3 Unterrichtsprojekt in einem Grundkurs am Fürst-Pückler-Gymnasium, Cottbus

Im November 2002 führte F. RIEPER in einem Grundkurs ma-13 am Fürst-Pückler-Gymnasium in Cottbus ein dreiwöchiges Unterrichtsprojekt durch, in dem sich die Schüler mit der Koordinatenbeschreibung geometrischer Objekte und der Erstellung von 3D-Computergrafiken und -animationen mithilfe von POV-Ray beschäftigten. Obwohl das Projekt innerhalb des Stoffgebietes Analytische Geometrie stattfand, bezogen sich die hauptsächlich bearbeiteten Inhalte nicht unmittelbar auf die Rahmenplanvorgaben; vielmehr hatten die Schüler recht große Freiräume bei der Wahl der Themenschwerpunkte.<sup>27</sup> Das von RIEPER konzipierte und durchgeführte „Mathe-Projekt: Raytracing“ umfasste drei Phasen:

1. Einarbeitung in POV-Ray, Modellierung geometrischer Objekte mithilfe von Koordinaten;
2. Bearbeitung eines jeweils größeren Projektes durch die Schüler in Einzel- oder Gruppenarbeit (hauptsächlich zu Hause);
3. Präsentation und Besprechung der Schülerprojekte im Unterricht.

In der ersten Projektphase bearbeiteten die Schüler anhand von Beispieldateien kleinere Aufgaben zu folgenden Themengebieten:

- Beschreibung von Grundkörpern (Kugeln, Quader, Zylinder, Kegel und Tori),
- Boolesche Operationen (Vereinigungen, Differenzen, Durchschnitte) von Körpern,
- Generieren komplexerer Objekte (Prismen und vor allem Rotationskörper),<sup>28</sup>
- Anfertigung eines einfachen Videos.<sup>29</sup>

Der Lehrer stellte dazu ein Anleitungsmaterial und Beispieldateien zur Verfügung, welche die Schüler zunächst nutzen und variieren konnten.<sup>30</sup> Damit erhielten sie einen ersten Einblick in die Möglichkeiten und die Arbeitsweise bei der Erstellung von Computergrafiken und -animationen mit POV-Ray. Auf dieser Grundlage konnten die Schüler bereits über die Inhalte ihrer Projekte nachdenken.

<sup>27</sup>Der Rahmenplan von Brandenburg beinhaltet u. a. die „Bearbeitung komplexer Problemstellungen“ mit recht offen gehaltenen Vorgaben. In diesem Rahmen sind auch Unterrichtsprojekte möglich, die sich nicht unmittelbar auf die obligatorischen Lehrplaninhalte beziehen.

<sup>28</sup>Auf die Erstellung von Rotationskörpern (siehe auch Abschnitt 2.3.7 der vorliegenden Arbeit) wurde im Unterricht etwas ausführlicher eingegangen. Obwohl keine Beschreibung durch Gleichungen, sondern durch Stützpunkte rotierender Kurven erfolgte, beabsichtigte RIEPER damit u. a., zumindest anhand der betrachteten Objekte Bezüge zum vorangegangenen Stoffgebiet Integralrechnung herzustellen, in welchem Rotationskörper und deren Volumina recht ausführlich behandelt wurden.

<sup>29</sup>Im Zusammenhang mit Tori und Booleschen Operationen sowie bei der Erstellung des Videos wurde auch kurz auf Beschreibungen von Verschiebungen durch Translationsvektoren und von Drehungen durch die Drehwinkel um die Koordinatenachsen eingegangen.

<sup>30</sup>Das Anleitungsmaterial mit den zugehörigen Beispielen befindet sich in dem Ordner `Schuelerarbeiten\Projekt-FPG-Cottbus-ma13-F-Rieper\Anleitung+Beispiele` (siehe S. 365) und steht auch unter <http://fpg-cottbus.de/faecher/mathematik/povrayprojekt.html> zur Verfügung.

Für die zweite Projektphase, die den größten Teil der Zeit einnahm und in der die Schüler größere Projekte bearbeiteten, wurde ihnen folgende Aufgabe gestellt:

*Bearbeiten Sie eine komplexe Aufgabe. Programmieren Sie möglichst effektiv mit Hilfe von Vereinigungen, damit sie Gruppen von Elementen manipulieren können (und nicht jedes Objekt einzeln). Notieren Sie Schwierigkeiten und Herangehensweisen für eine Präsentation.*

**Vorschläge**

1. **Stilleben:** Erstellen Sie ein Stilleben bestehend aus einem Tisch mit realen Gefäßen und geometrischen Formen (Rotationskörpern).
2. **Kegelschnitte:** Schneiden Sie einen Kegel mit einer Ebene und stellen Sie die verschiedenen Schnittmengen übersichtlich dar.
3. **Technik:** Konstruieren Sie ein Fahrzeug oder anderes technisches Gerät (Zahnräder) und erzeugen Sie dazu eine Animation.

Die Vorschläge waren tatsächlich als solche aufzufassen und nicht als Vorgaben; dies zeigte sich auch in den Projekten einiger Schüler, die sich andere Schwerpunkte setzten, z. B. die Anfertigung von Animationen anderen als des vorgeschlagenen Inhalts durch die zeitabhängige Beschreibung von Verschiebungen und Drehungen. Hinsichtlich der Präsentation wurde im Verlauf des Unterrichts noch herausgestellt, dass die Schüler über mindestens ein *geometrisches* Problem vortragen sollten, das bei der Arbeit auftrat, da technische Aspekte nicht zu stark in den Vordergrund treten sollten.

Die von den Schülern angefertigten Projekte unterscheiden sich hinsichtlich ihrer Komplexität und der zugrunde liegenden mathematischen Beschreibungen erheblich. Die Spannweite reicht von der Anordnung einfacher geometrischer Objekte (verbunden mit der Zuweisung vorgefertigter Texturen und der Erstellung einfacher Animationen durch die zeitabhängige Beschreibung eines Drehwinkels) wie in dem Projekt **Frank-Doom** bis hin zu ausgereiften Modellierungen realer Objekte durch Rotationskörper und der Generierung komplexer Animationen in dem Projekt **Juliane-Mittelalter**.<sup>31</sup> Im Folgenden wird eine Zusammenfassung der von den Schülern in ihren Arbeiten genutzten mathematischen Inhalte gegeben.

- Viele Schüler verwendeten *Grundkörper*, *Prismen* und *Boolesche Operationen*, um Objekte zu modellieren (siehe u. a. **Martin-Leuchtturm**, **Stefanie-Weltraum**, **SteffenL-Lord-of-Onions**, **Raymond-Kugeln**, **Undine-Wanduhr**, **Wenke-Kathi-Zimmer**). Zum Teil entstanden dabei recht aufwändige Szenen, wobei Unterschiede zwischen den einzelnen Schülern auftraten, die mit den Komplexitätsunterschieden der in den vorangegangenen Abschnitten beschriebenen Schneemänner vergleichbar sind. Allerdings ergänzten die meisten Schüler in dem hier skizzierten Unterrichtsprojekt ihre Szenen durch Animationen von Objekten.
- In den Projekten einiger Schüler nahmen *Rotationskörper* eine wichtige Stellung ein (**Juliane-Mittelalter**, **Katrin-Stilleben**, **Wenke-Kathi-Zimmer**).

<sup>31</sup>Die von den Schülern bzw. Schülergruppen angefertigten Arbeiten befinden sich in dem Ordner `Schuelerarbeiten\Projekt-FPG-Cottbus-ma13-F-Rieper`; einige Abbildungen daraus sind auf S. 361f. wiedergegeben.

- Neun von elf Schülern bzw. Schülergruppen fertigten *Animationen* an, die zum größten Teil durch die zeitabhängige Darstellung von Drehwinkeln realisiert wurden. Eine Schülerin beschäftigte sich mit *Nacheinanderausführungen von Bewegungen*, wodurch sie Drehungen (recht abstrakt erscheinender) Himmelskörper um ihre eigene Achse sowie um andere Objekte realisierte (**Stefanie-Weltraum**).
- Eine hinsichtlich der mathematischen Beschreibung etwas kompliziertere Animation fertigten zwei Schülerinnen an, indem sie die Bewegung des Pendels einer Uhr darstellten. Um diese *periodische Bewegung* zu realisieren, drückten sie den Drehwinkel als Sinus des Zeitparameters aus (**Undine-Wanduhr**).
- In drei Projekten wurden *Parameterdarstellungen von Kurven* verwendet, um Animationen zu generieren.<sup>32</sup> Eine Schülerin beschrieb eine elliptische Bahn (auf der sich eine „Erde“ um eine „Sonne“ bewegt, siehe **Mandy-Erde-und-Sonne**) in der winkelhalbierenden Ebene zwischen der  $x$ - $y$ - und der  $y$ - $z$ -Ebene durch  
 $\langle 6 \cdot \sin(\text{clock} \cdot 2 \cdot \pi), 6 \cdot \cos(\text{clock} \cdot 2 \cdot \pi), 6 \cdot \sin(\text{clock} \cdot 2 \cdot \pi) \rangle$ .

Für die Animation zweier Scheinwerfer beschrieb ein Schüler Kreise, auf denen sich deren Ziele bewegen, durch Parameterdarstellungen (**Martin-Leuchtturm**):

```
light_source { <20,60,15> color White
  spotlight radius 10 falloff 20 tightness 5
  point_at <-50*sin((3.14*clock+20)/180),1,-50*cos((3.14*clock+20)/180)>}
```

In dem bereits erwähnten Projekt **Juliane-Mittelalter** animierte die Schülerin mehrere Objekte, wobei sie sowohl zeitabhängige Transformationen als auch durch Parameterdarstellungen beschriebene Kurven nutzte, u. a. für den Lichtkegel eines Scheinwerfers, dessen Zielpunkt sich auf einer Lissajous-Kurve bewegt:

```
point_at <150*sin(clock/15), 40*cos(clock/30)-20, 60>.33
```

Noch komplexere Beschreibungen liegen dem Flackern der Kerzen in dem Video dieser Schülerin zugrunde, das durch zeitlich versetzte Drehungen von Rotationskörpern mit sich periodisch ändernder Winkelgeschwindigkeit entstand.<sup>34</sup>

- Zwei Schüler beschäftigten sich in ihrem (gemeinsamen) Projekt mit der *Darstellung von Kegelschnitten* (**Tobias-Steffen-Kegelschnitte**). Ihre Arbeit konzentrierte sich auf die dreidimensionale Darstellung der Schnittfiguren, wobei sie die Entstehung einer Hyperbel, eines Kreises und einer „echten“ Ellipse (u. a. durch ein einfaches Video) veranschaulichten; Parabeln blieben jedoch unberücksichtigt.

<sup>32</sup>Parameterdarstellungen von Kurven wurden zuvor im Unterricht nicht besprochen und waren auch nicht in der Anleitung enthalten, die den Schülern zur Verfügung stand. Die betreffenden Schüler sahen durch die Nutzung von zeitabhängigen geometrischen Transformationen keine Möglichkeit, ihre Vorstellungen von den Animationen, die sie erstellen wollten, zu realisieren und kamen dadurch dazu, sich mit Parameterdarstellungen zu beschäftigen. Dabei wurden sie vom Lehrer des Kurses unterstützt.

<sup>33</sup>Die entsprechende Animation findet sich im Vorspann des von der Schülerin hergestellten Videos **The-Middleage-Video.mpg**, das sich aus vier Teilvideos zusammensetzt, die sie durch die POV-Ray-Dateien **Intro.pov**, **Mittelalterliche\_Tafel.pov**, **Schwert.pov** und **Outro.pov** erzeugte.

<sup>34</sup>Diesen Effekt erreichte die Schülerin, indem sie die Drehwinkel durch Sinus- bzw. Kosinusfunktionen des Zeitparameters ausdrückte, siehe die Zeilen 330–402 der Datei **Mittelalterliche\_Tafel.pov**.

Zusammenfassend sind m. E. folgende Ergebnisse des Unterrichtsprojektes hervorhebenswert:

- Alle Schüler erreichten zumindest akzeptable Resultate, die dem Ziel, geometrische Objekte durch Koordinaten zu beschreiben und im Raum anzuordnen, entsprachen.
- Die Unterschiede in der Komplexität der modellierten Szenen und der Vielfalt der verwendeten Objekte waren enorm. Einige Schüler erstellten Szenen, die im Hinblick auf
  - die Modellierung von Objekten (Boolesche Operationen, Rotationskörper),
  - die verwendeten geometrische Transformationen (insbesondere bezüglich der Wahl der Drehwinkel) sowie
  - die genutzten Techniken der Animation (zeitabhängige Beschreibung von Transformationen, Parameterdarstellungen von Bahnkurven)
 sehr anspruchsvoll sind.<sup>35</sup>
- Es kamen hauptsächlich elementare mathematische Beschreibungen zur Anwendung, die aber zum großen Teil von den Schülern selbstständig erarbeitet wurden.
- Als besonders bemerkenswert ist anzusehen, dass einige Schüler durch die Arbeit an ihren Projekten motiviert wurden, sich mit Parameterdarstellungen von Kurven zu befassen und dies auch erfolgreich taten.<sup>36</sup>
- Die Zeit, die viele Schüler freiwillig in häuslicher Arbeit mit ihren Projekten verbrachten, überstieg bei weitem das für Schüler von Grundkursen übliche Maß.
- Den Schülern bereitete die Beschäftigung mit der Aufgabe viel Freude. Als besonders bedeutsam für die erreichte hohe Motivation erwiesen sich die optisch attraktiven Ergebnisse sowie der bei vielen Schülern vorhandene Wunsch, Videos zu erstellen.

---

<sup>35</sup>Es könnte der Verdacht geäußert werden, dass einige Schüler Dateien, die sie z. B. auf einschlägigen Internetseiten fanden, lediglich variierten. Eine recht umfangreiche Recherche entsprechender Quellen führte jedoch zu dem Ergebnis, dass derartige Fälle höchstwahrscheinlich nicht auftraten. Einige Schüler verwendeten zwar offenbar einzelne Elemente (z. B. Wasseroberflächen, Himmel und ein Uhrenpendel), die sie insbesondere auf der (in der Anleitung des Lehrers empfohlenen) Seite von LOHMÜLLER [http://www.f-lohmueller.de/pov\\_tut/pov\\_\\_ger.htm](http://www.f-lohmueller.de/pov_tut/pov__ger.htm) gefunden hatten, nutzten diese jedoch lediglich für die Ergänzung ihrer eigenen Arbeiten. In derartiger Weise Anregungen aufzugreifen und auch vorliegende Objekte in eigene Arbeiten zu integrieren, ist ein allgemein übliches und legitimes Vorgehen; Plagiate fertigten die Schüler damit nicht an. Hinsichtlich des in jeder Hinsicht herausragenden Projektes **Juliane-Mittelalter** sei angemerkt, dass sich keine dazu vergleichbaren Arbeiten finden ließen. Auch wenn die Schülerin fremde Hilfe nutzte (der abschließende Videoschnitt und die Vertonung wurden von ihrem Freund vorgenommen), mindert dies die von ihr erbrachte Leistung nicht.

<sup>36</sup>Die Vertiefung dieser Thematik im Anschluss an die Präsentation der Projekte wäre m. E. besonders lohnenswert gewesen, wenn genügend Zeit zur Verfügung gestanden hätte. Mögliche Richtungen und Vorgehensweisen wurden in Abschnitt 4.3 beschrieben. In der hier gegebenen Situation hätte es sich vor allem angeboten, diejenigen Schüler, die in ihren Projekten Parameterdarstellungen verwendeten, ihre Erfahrungen weitergeben zu lassen und darauf aufbauend vertiefende Überlegungen anzustellen. Auf dieser Grundlage hätten die Schüler dann weitere Animationen erstellen oder interessante Kurven, evtl. sogar auch Flächen, darstellen können.

## 5.4 Erfahrungen aus Seminaren mit Studierenden

Auch wenn sich die Arbeit mit Studierenden nicht in die Thematik der vorliegenden Arbeit einordnen lässt, wird im Folgenden kurz auf Erfahrungen eingegangen, die in zwei (wahlobligatorischen) Seminaren zur geometrischen Modellierung in der 3D-Computergrafik<sup>37</sup> gesammelt wurden. Einige Aspekte, die bereits bei dem in Abschnitt 5.3 beschriebenen Unterrichtsprojekt von Bedeutung waren, werden dadurch ergänzt, obwohl Unterschiede hinsichtlich der Ziele und Ausgangsbedingungen zu erwähnen sind. So stand das Erlernen von Inhalten der analytischen Geometrie<sup>38</sup> nicht im Mittelpunkt der Seminare; vielmehr sollten die Studierenden anspruchsvolle geometrische Modellierungen vornehmen, die auf der Koordinatengeometrie des Raumes, vor allem aber auf Inhalten des Mathematikunterrichts der SI (Kongruenz- und Ähnlichkeitsabbildungen, Raumgeometrie, trigonometrische Funktionen) basieren.

Die Seminare begannen mit einer Einführung in grundlegende Vorgehensweisen der 3D-Computergrafik,<sup>39</sup> an die sich die bereits beschriebene Modellierung eines Schneemannes anschloss.<sup>40</sup> Wie bei den Schülern zeigten sich enorme Unterschiede zwischen den Ergebnissen, wenngleich auf einem insgesamt höherem Niveau.<sup>41</sup> Im Anschluss erfolgten Einführungen in die Verwendung von Texturen, die Erstellung von Videos sowie die Modellierung mit Prismen, Booleschen Operationen und Rotationskörpern. Außerdem führten die Studierenden Experimente mit Translationen, Drehungen und Streckungen durch. Im Zusammenhang mit der Erstellung von Animationen wurde auch auf die Beschreibung von Bahnkurven durch Parameterdarstellungen eingegangen.<sup>42</sup>

<sup>37</sup>Die Seminare fanden im Sommersemester 2004 sowie im Wintersemester 2004/05 an der Pädagogischen Hochschule Heidelberg statt. Es nahmen Studierende für das Mathematik-Lehramt an Grund-, Haupt- und Realschulen teil.

<sup>38</sup>Die Studierenden hatten wenige Jahre zuvor als Schüler die Rahmenplaninhalte des Stoffgebietes Analytische Geometrie kennengelernt; allerdings war festzustellen, dass die Mehrzahl von ihnen nur noch über sehr rudimentäre Kenntnisse und Fähigkeiten verfügte, siehe auch die Fußnote 66 auf S. 34.

<sup>39</sup>In dem Seminar im Wintersemester 2004/05 wurden darüber hinaus am Anfang zwei Sitzungen für Aspekte der Bildbearbeitung genutzt. Zu den Inhalten gehörten dabei u. a. die Beschreibung von Farben durch RGB-Tripel sowie Übungen zur Bildbearbeitung mithilfe der Software Adobe Photoshop.

<sup>40</sup>Dabei wurde im Vergleich zu den zeitlich wesentlich enger begrenzten Unterrichtsreihen an Schulen verstärkt Wert auf die Herleitung und Nutzung der Kreis- und Kugelgleichungen gelegt. Ein Ergebnis eines Studenten, der eine ganze „Schneemannarmee“ modellierte, wurde bereits auf S. 258f. kurz erwähnt (Ordner: `Studierenden-Arbeiten\Schlager_Schneemannarmee`). Anzumerken ist dazu, dass der betreffende Student als einziger Seminarteilnehmer in seiner Schulzeit Informatikkurse belegt hatte.

<sup>41</sup>Die von Studierenden angefertigten POV-Ray-Dateien und die daraus berechneten Bilder sowie Videos befinden sich in der elektronischen Materialsammlung zu der vorliegenden Arbeit (siehe S. 365) sowie auf der Internetseite <http://www.ph-heidelberg.de/wp/filler/lv-alt/compgraf/ergebn-stud.html>. Die Studierenden haben einer Veröffentlichung ausdrücklich zugestimmt.

<sup>42</sup>In einem der Seminare führten die Studierenden dazu Übungen durch, wobei sie sowohl Objektanimationen auf kreis- und spiralförmigen Bahnen generierten, als auch die entsprechenden Kurven mithilfe von `while`-Schleifen durch Kugeln darstellten (siehe S. 152 und S. 249ff.). Die Ergebnisse einiger der von den Studierenden vorgenommenen Experimente befinden sich in dem Ordner `Studierenden-Arbeiten\Bewegungsbahnen`



Auf der Grundlage der skizzierten Überlegungen arbeiteten die Studierenden einzeln oder in Gruppen an größeren Projekten. Dabei wurde die Anforderung gestellt, komplexere Szenen unter anspruchsvollen Modellierungen geometrischer Objekte zu erstellen oder interessante Animationen durch Parameterbeschreibungen von Animationskurven oder Nacheinanderausführungen mehrerer zeitabhängig beschriebener Transformationen zu generieren. Jeweils in den letzten beiden Seminaren stellten die Studierenden ihre Projekte vor, wobei Wert auf die Beschreibung mathematischer Aspekte der Arbeiten zu legen war. Ohne die Projekte der Studierenden detailliert zu beschreiben, seien zusammenfassend folgende Ergebnisse genannt:<sup>43</sup>

- Die Projekte unterschieden sich im Hinblick auf ihre Komplexität und das mathematische Niveau der verwendeten Beschreibungen erheblich; die weitaus meisten Arbeiten belegen jedoch, dass sich die entsprechenden Studierenden sorgfältig mit ihren Projekten beschäftigt und fundierte Überlegungen hinsichtlich der Modellierung von Objekten und der Beschreibung von Animationen angestellt haben.
- Die Bildinhalte und Herangehensweisen der Studierenden weisen eine große Breite auf, wie bereits ein flüchtiger Blick auf die beiden Internetseiten verdeutlicht.
- Die meisten Studierenden wählten als Schwerpunkt ihres Projektes komplexere geometrische Modellierungen,<sup>44</sup> wobei in vielen Fällen Rotationskörper unterschiedlichster Art als dominierende Gestaltungselemente zum Einsatz kamen. Exemplarisch sei das Projekt **Engelland\_Schachspiel** angeführt, zu dem sich auch eine kurze Beschreibung in dem entsprechenden Ordner befindet.
- Einige Studierende wählten anspruchsvollere Animationen als Projektschwerpunkt und nutzten Parameterdarstellungen von Kurven für Kameraanimationen.
- Ein Beispiel für eine geometrisch recht komplexe Szene, die zugleich interessante (durch zeitabhängige Beschreibungen von Bewegungen generierte) Animationen enthält, ist das Projekt **Lipp+Hainbuch\_MenschAerger**.
- Die Studierenden hatten Freude an der Anfertigung ihrer Projekte und investierten recht viel Zeit in ihre Arbeiten. In überdurchschnittlich hohem Maße gelang es in den Seminaren, auch Studierende, die ansonsten eher zu rezeptivem Lernen und der Abarbeitung klar abgegrenzter Aufgaben neigten, zu einem aktiven, selbstgesteuerten Arbeiten zu motivieren. In der Kurzbeschreibung ihres Projektes führte eine Studentin aus: „Für mich war das Seminar zwar das Zeitaufwendigste, aber zugleich auch das erste, das mich etwas mehr gefordert hat...“<sup>45</sup>

---

<sup>43</sup>Siehe dazu auch die Dateien in dem Ordner **Studierenden-Arbeiten**.

<sup>44</sup>Zum Teil ergänzten diese Studierenden ihre Szenen durch einfachere Animationen, die jedoch nicht den Schwerpunkt der Arbeiten bildeten.

<sup>45</sup>Die Kurzbeschreibung befindet sich in dem Ordner **Studierenden-Arbeiten\Kara\_SoupFor2**. Auch einige Projektbeschreibungen anderer Studierender enthalten interessante Einschätzungen zu ihrer Arbeit in dem Seminar.

## 5.5 Abschließende Bemerkungen

Die in den vorangegangenen Abschnitten beschriebenen Unterrichtsversuche reichten von der Ergänzung „traditioneller“, eng vorgegebener Inhalte einer vektoriellen Geometrie des Raumes durch Computervisualisierungen und die Behandlung einzelner Elemente der Computergrafik bis hin zu einem recht frei gestalteten Unterrichtsprojekt, in dessen Mittelpunkt Modellierung und Animation in einer Grafiksoftware standen. Trotz dieser unterschiedlichen Ansätze der Einbeziehung von Elementen der Computergrafik lassen sich einige übergreifende Aussagen treffen:

- In allen Unterrichtsversuchen gelang es, bei der Mehrheit der Schüler durch die Arbeit mit einer 3D-Grafiksoftware ein hohes Maß an *Motivation* zu erreichen und *Neugier* zu wecken. Dies führte dazu, dass viele Schüler auch in der Freizeit an ihren Computergrafiken arbeiteten.
- Die Erstellung und Perfektionierung von 3D-Grafiken durch die Schüler hat mindestens dazu geführt, dass sie eine gewisse „Orientierungsfähigkeit“ im räumlichen Koordinatensystem bzw. *koordinatenbezogene Raumvorstellung* (vgl. 1.4.4) erworben haben, ohne die dreidimensionale Szenen in einer Software wie POV-Ray nicht beschrieben werden können.
- Das *anschauliche Verständnis der behandelten Gegenstände des Stoffgebietes Analytische Geometrie* wird durch Visualisierungen in beachtlichem Maße gefördert. Damit ist noch nicht zwangsläufig eines tieferes Verständnis für die Tragfähigkeit und Verallgemeinerbarkeit von Objekten und Methoden der analytischen Geometrie erreicht (das sich allein durch Veranschaulichungen auch nicht erreichen lässt). Gegenüber einem weitgehend auf die Abarbeitung von Kalkülen beschränkten Unterricht lässt die Ergänzung durch grafische Darstellungen, welche die Schüler selbst erstellen, geometrische Inhalte und anschauliche Vorstellungen stärker in den Vordergrund treten. Dies ist m. E. selbst dann als Bereicherung anzusehen, wenn die untersuchten geometrischen Objekte recht uninteressant sind, es sich also (wie nach vielen gegenwärtigen Rahmenplänen vorgesehen) fast nur um Geraden und Ebenen handelt – unbefriedigend bleibt diese Beschränkung dennoch.
- Die Arbeit mit relativ *offen gehaltenen Aufgabenstellungen* wie der Modellierung eines Schneemanns oder den in Abschnitt 5.3 skizzierten Projektaufträgen erwies sich in mehrfacher Hinsicht als sinnvoll. So sind für alle Schüler Erfolgserlebnisse möglich und auch Schüler mit geringer ausgeprägten mathematischen Fähigkeiten oder nur rudimentären Kenntnissen hinsichtlich der Nutzung des Computers können Freude an der Arbeit empfinden. Gleichzeitig bestehen für besonders interessierte und leistungsfähige Schüler im Rahmen derartiger Aufgaben Freiräume, um außergewöhnliche Ergebnisse zu erreichen.<sup>46</sup> Die von DE-

---

<sup>46</sup>Aufgaben, die Lösungen unterschiedlicher Schwierigkeitsgrade zulassen, dürften sich generell gut für Einstiege in Unterrichtsreihen eignen, in denen die Computernutzung einen hohen Stellenwert hat, da durch derartige Aufgaben Unterschiede der Schüler hinsichtlich des Umgangs mit dem Computer und die daraus resultierenden unterschiedlichen Einarbeitungszeiten ausgeglichen werden können.

CI und RYAN formulierte und durch empirische Untersuchungen untermauerte These, dass „Kompetenz- und Autonomieerfahrungen ... eine wichtige Bedingung darstellen, um intrinsische Motivation herzustellen und aufrechtzuerhalten“ ([46], S. 232) bestätigte sich bei den hier beschriebenen Unterrichtsreihen.<sup>47</sup> Es war im Unterricht und bei Gesprächen mit Schülern erfahrbar, dass viele von ihnen mit Stolz auf ihre Arbeiten blickten. Neben dem Gegenstand und den Produkten der 3D-Computergrafik dürften somit auch prozessbezogene Aspekte, insbesondere Autonomie- und Kompetenzerlebnisse der Schüler bei der selbstständigen Bearbeitung von Aufgaben, die Freiräume bei der Lösung ließen, zu der bereits beschriebenen hohen Motivation vieler Schüler beigetragen haben.

- Die Arbeit der Schüler mit einer Grafiksoftware an relativ offen gehaltenen Aufgaben führte in allen Unterrichtsversuchen dazu, dass sich viele Schüler, motiviert durch den Wunsch, interessante Gestaltungsmöglichkeiten für ihre Grafiken zu erschließen, *freiwillig mit mathematischen Inhalten beschäftigten*. Beispiele hierfür sind elementare Beschreibungen geometrischer Transformationen sowie Parameterdarstellungen von Kurven, die für die Erstellung von Animationen benötigt werden. Insbesondere das zuletzt genannte – hinsichtlich der Herausbildung zentraler mathematischer Ideen äußerst ertragreiche – Themengebiet kann durch die Arbeit mit einer 3D-Grafiksoftware sehr gut motiviert werden.
- Mitunter geäußerte Bedenken, die Arbeit mit einer Skript- bzw. Szenenbeschreibungssprache (SDL) könnte viele Schüler überfordern, erwiesen sich aufgrund aller gesammelten Erfahrungen als unbegründet. Technische Probleme bzw. Probleme mit der Syntax der POV-Ray-SDL wirkten sich kaum problematisch aus und kosteten nur wenig Unterrichtszeit. Eine wichtige Voraussetzung hierfür war sicherlich, dass den Schülern geeignete Anleitungen und Vorlagen bzw. Beispieldateien zur Verfügung gestellt wurden.<sup>48</sup>

---

<sup>47</sup>Als weitere Bedingung dafür, dass eine Aktivität intrinsisch motiviert vollzogen wird, führten DECI/RYAN an, dass diese ein „für das Individuum optimales Anforderungsniveau“ besitzen sollte. „Dies ist der Fall, wenn zwischen den Anforderungen einer zielbezogenen Tätigkeit und dem aktuell gegebenen Fähigkeitsniveau eine optimale Diskrepanz besteht und die zu bewältigende Aufgabe weder als zu leicht noch als zu schwer empfunden wird“ ([46], S. 231). Mit den gestellten Aufgaben, die Lösungen unterschiedlichen Schwierigkeitsgrades zuließen, gelang es offenbar, für einen sehr großen Teil der beteiligten Schüler angemessene Anforderungsniveaus zu erreichen.

<sup>48</sup>Bei einigen im Umgang mit dem Computer unerfahrenen Schülern waren die Hauptprobleme bei der Computernutzung sehr allgemeiner Art und bestanden zum Beispiel im Speichern und Auffinden von Dateien. Derartige Probleme wären auch bei der Nutzung jeder anderen Software aufgetreten; sie konnten recht leicht gelöst werden, wofür sich, wie bereits erwähnt wurde, der Einstieg über relativ offene Aufgaben als sinnvoll erwies. Aufgrund von Beobachtungen konnte sogar der Eindruck gewonnen werden, dass die Nutzung einer Software, mit der vor allem durch Eingaben in ein Editorfenster gearbeitet wird, gerade für unerfahrene Schüler günstig war. Hatten sie POV-Ray erst einmal gestartet und waren in der Lage, Dateien zu öffnen und zu speichern, so benötigten sie keine „technischen Tricks“ mehr, um mit der Software zu arbeiten, sondern mussten vor allem über die richtige Beschreibung von Objekten durch Koordinaten nachdenken. Hierbei waren sie gegenüber den im Umgang mit dem Computer versierteren Schülern nicht mehr benachteiligt. Software, die vor allem mit der Maus und über komplexe Menüstrukturen zu bedienen ist, könnte in dieser Hinsicht problematischer sein.

Die folgenden Beobachtungen und Schlussfolgerungen beziehen sich nicht speziell auf die Einbeziehung von Elementen der Computergrafik in das Stoffgebiet Analytische Geometrie, sondern könnten allgemein auf die Computernutzung im Mathematikunterricht zutreffen (vgl. Abschnitt 1.5), wenn Schüler an Aufgaben arbeiten, die ihnen ausreichende Freiräume lassen, um eigene Ideen umzusetzen. Sie seien hier dennoch aufgeführt, da sie in den beschriebenen Unterrichtsversuchen die Wahrnehmung der unterrichtenden Lehrer und die Einschätzung der Schüler wesentlich bestimmten:

- Bei der Anfertigung von Grafiken mithilfe von POV-Ray kam es in überdurchschnittlich hohem Maße sowohl zu einer Kooperation zwischen Schülern innerhalb von Gruppen, die gemeinsam an Grafiken arbeiteten, als auch zu einem Austausch zwischen Gruppen. Dabei erhaltene Anregungen griffen Schüler zwar häufig auf, „kopierten“ jedoch nicht einfach Ideen anderer Schüler bzw. Schülergruppen, sondern variierten sie und integrierten sie auf diese Weise in ihre eigenen Arbeiten, wie anhand der auf S. 357ff. abgebildeten Grafiken sichtbar wird.
- Während selbstständiger Schülerarbeitsphasen traten die Lehrer in starkem Maße als „Berater“ in Erscheinung, denen die Schüler Fragen zu technischen Problemen, vor allem aber zu mathematischen Inhalten (insbesondere bezüglich der Erstellung von Animationen) stellten.<sup>49</sup> Situationen, in denen Schüler z. B. den Wunsch äußern, Parameterbeschreibungen von Bahnkurven zu behandeln, gehören ansonsten nicht immer zum Unterrichtsalltag und stellten für die unterrichtenden Lehrer interessante und erfreuliche Erfahrungen dar.
- Wesentlich mehr als im sonstigen Unterricht wurden die Lehrer von den Schülern auf Aspekte und Ideen aufmerksam gemacht, die auch für die betreffenden Lehrer neu waren. Sie konnten somit selbst von den Schülern „etwas lernen“. Auf die Motivation und das Selbstbewusstsein der Schüler hatte dies einen positiven Einfluss. Die bereits auf S. 59 zitierte Bemerkung „*Erfahrungen zum Computereinsatz geben heute zu der Hoffnung Anlass, dass neue Technologien ein Katalysator für eine ... ,neue Unterrichtskultur‘ sein können*“ ([29], S. 86) kann aufgrund der in diesem Kapitel beschriebenen Unterrichtsversuche bestätigt werden.

Bei den beschriebenen Untersuchungen standen geschlechtsspezifische Aspekte nicht im Mittelpunkt und wurden nicht explizit erfasst und ausgewertet. Dennoch ergibt sich aus allen gesammelten Erfahrungen eine Aussage, die gerade im Hinblick darauf, dass mitunter Benachteiligungen bzw. besondere Probleme von Mädchen bei der Nutzung von Computern konstatiert oder zumindest befürchtet werden, interessant erscheint:

- In keinem der durchgeführten Unterrichtsversuche ließ sich feststellen, dass Mädchen bzw. junge Frauen größere Schwierigkeiten bei der Arbeit mit POV-Ray hatten als Jungen oder bei ihnen eine größere Hemmschwelle gegenüber der Arbeit am PC bestand. Die von Mädchen erzielten Ergebnisse sind – zusammenfassend

<sup>49</sup>Wie bereits erwähnt wurde, wendeten sich Schüler auch in Pausen häufig mit Fragen zu der Arbeit an ihren Grafiken an die Lehrer. Einige Stunden der in Abschnitt 5.3 beschriebenen Unterrichtsreihe hatten den Charakter von Konsultationen, in denen bei den Projekten der Schüler aufgetretene Probleme technischer, vor allem aber mathematischer Natur besprochen wurden.

betrachtet – gegenüber denen der Jungen als gleichwertig anzusehen. In den beiden Unterrichtsversuchen, die in den Abschnitten 5.1 und 5.3 beschrieben wurden, fertigten Mädchen die jeweils herausragenden Arbeiten an. Ein gewisser Unterschied zwischen Jungen und Mädchen konnte dahin gehend beobachtet werden, dass Jungen oft etwas „verspielter“ und in Bezug auf bestimmte Belange (wie z. B. Texturierungen) experimentierfreudiger arbeiteten als Mädchen, welche wiederum ein höheres Maß an Zielorientierung aufwiesen und in vielen Fällen künstlerisch überzeugendere Grafiken gestalteten. Diese Aussage gibt aber nur eine Tendenz wieder und trifft nicht auf alle beteiligten Schülerinnen und Schüler zu.

Sollte es tatsächlich der Fall sein, dass Mädchen bei der Arbeit mit dem Computer im Allgemeinen größere Schwierigkeiten haben als Jungen, so könnte die Tatsache, dass dies bei keiner der beschriebenen Unterrichtsreihen zu erkennen war, darauf zurückzuführen sein, dass für die Computergrafik auch künstlerische Aspekte von Bedeutung sind. Kunst gehört zu den Lieblingsfächern vieler Mädchen, was dazu führen könnte, dass diese einer künstlerische Momente beinhaltenden Nutzung des Computers besonders aufgeschlossen gegenüberstehen.<sup>50</sup> Es handelt sich bei dieser Hypothese lediglich um eine Mutmaßung, die nicht näher untersucht wurde. Als Tatsache bleibt festzuhalten, dass in Kursen mit insgesamt ca. 60 Schülern sowie in Seminaren mit ca. 30 Studierenden Mädchen bzw. junge Frauen insgesamt in keiner Hinsicht schlechtere Ergebnisse erreichten als junge Männer.

Alle genannten Ergebnisse und Erfahrungen zusammenfassend ist m. E. zu konstatieren, dass die Einbeziehung von Elementen der Computergrafik in das Stoffgebiet Analytische Geometrie und die Arbeit der Schüler an umfangreicheren, recht offen gestellten Aufgaben unter Verwendung einer Grafiksoftware natürlich Unterrichtszeit benötigen, die aber damit in sinnvoller Weise investiert wird.

### Einige Konsequenzen hinsichtlich der Gestaltung künftiger Curricula

Wie bereits erwähnt wurde, konnten nicht alle der in Kapitel 4 unterbreiteten Vorschläge in Unterrichtsversuchen umgesetzt werden; einen wichtigen Grund hierfür stellten die Vorgaben der Rahmenpläne dar, nach denen das Stoffgebiet zu unterrichten war. Diese Vorgaben führten auch dazu, dass im Unterricht Brüche erfolgten sowie Themengebiete, an denen die Schüler Interesse zeigten und die auch inhaltlich ergiebig gewesen wären, nicht weitergeführt werden konnten. Bereits in Abschnitt 4.9 wurden einige Richtungen skizziert, hinsichtlich derer inhaltliche Vertiefungen anstrebenswert wären. Aufgrund der gesammelten Erfahrungen ergeben sich zwei Aspekte, deren Berücksichtigung bei künftigen Überarbeitungen von Rahmenplänen besonders sinnvoll erscheint:

- Wünschenswert sind größere Freiräume für Unterrichtsprojekte, deren Schwerpunkte (die sich natürlich auf ergiebige mathematische Ideen und Inhalte bezie-

---

<sup>50</sup>So belegte die Schülerin, welche den auf S. 229f. beschriebenen und in Abb. B.30 (rechts) auf S. 358 farbig dargestellten Schneemann anfertigte, einen Kunst-Leistungskurs, was nach ihren eigenen Angaben Einfluss auf die Gestaltung der Grafik hatte.

hen müssen) Lehrer und Schüler bestimmen können. Dazu ist eine Verringerung des Anteils zu behandelnder „Aufgabeninseln“ anzustreben, siehe Abschnitt 1.4.2 und die diesbezüglichen Bemerkungen auf S. 313f.

- Die keineswegs neue Forderung, interessante geometrische Objekte im Unterricht der analytischen Geometrie zu behandeln und die weitgehende Beschränkung auf Geraden und Ebenen zu überwinden (vgl. 1.4.5), wurde durch die in der vorliegenden Arbeit angestellten Überlegungen und die gesammelten Erfahrungen noch erhärtet. Als besonders interessant erweist sich im Zusammenhang mit der Einbeziehung von Elementen der Computergrafik und -animation die Behandlung von Parameterdarstellungen von Kurven. Dieser Thematik stehen viele Schüler im Zusammenhang mit der Anfertigung von Animationen sehr aufgeschlossen gegenüber; gleichzeitig handelt es sich um einen Gegenstandsbereich, anhand dessen zentrale Ideen der Mathematik (wie die des funktionalen Zusammenhangs) sehr gut herausgearbeitet werden können. Wie in Abschnitt 4.3 dargestellt wurde, lassen sich Parameterdarstellungen von Kurven ansatzweise recht elementar, ausgehend von Inhalten des Mathematikunterrichts der SI, behandeln und ermöglichen bereits auf dieser Grundlage attraktive Anwendungen in der Computergrafik. Gleichzeitig lässt diese Thematik eine Weiterführung hin zu einem sehr anspruchsvollen mathematischen Niveau zu. Sowohl vertiefte Überlegungen zu Parameterdarstellungen von Kurven (bis hin zu Ansätzen der Differenzialgeometrie, siehe 4.3.3) als auch die Erarbeitung von Parameterdarstellungen spezieller Flächen (vgl. 4.5.4) erscheinen in leistungsstarken Kursen denkbar.<sup>51</sup>

Die Einbeziehung von Elementen der Computergrafik kann bereits unter den dafür recht einengenden Vorgaben der derzeit (noch) geltenden Rahmenpläne dazu beitragen, die Dominanz reinen Rechnens im Unterricht der analytischen Geometrie zu vermindern und das Stoffgebiet für Schüler attraktiver zu gestalten. Veränderungen der Rahmenbedingungen in die skizzierten Richtungen würden es ermöglichen, die Potenziale der Computergrafik hinsichtlich der Motivierung der Schüler, sich mit interessanten und anspruchsvollen mathematischen Inhalten zu beschäftigen, stärker zu nutzen.

---

<sup>51</sup>Die EPA (Einheitliche Prüfungsanforderungen in der Abiturprüfung) von 2002 und einige Tendenzen in neueren Rahmenplänen geben Anlass zu der Hoffnung, dass sich Veränderungen in die genannten Richtungen vollziehen. So sehen die EPA Kurven und gekrümmte Flächen im Raum zumindest als „möglichen zusätzlichen Themenbereich“ in Abiturprüfungen vor (vgl. [136], S. 8). Die weniger starren Vorgaben einiger in jüngster Zeit in Kraft getretener Rahmenpläne sowie Verbindungen zwischen Analysis bzw. zumindest Funktionenlehre und analytischer Geometrie (vgl. Abschnitt 1.3) könnten zu deutlich besseren Rahmenbedingungen für die Umsetzung in der vorliegenden Arbeit unterbreiteter Vorschläge führen. Über die konkrete Umsetzung der neuen Rahmenpläne in der Schulpraxis und Auswirkungen auf die Gestaltung der zentralen Abiturprüfungen können allerdings zum Zeitpunkt der Fertigstellung dieser Arbeit (Anfang 2006) noch keine Aussagen gemacht werden, da für die derzeitigen obersten Klassenstufen noch die älteren Rahmenpläne gelten.

# Anhang A

## Anleitungen für Schüler, Vorlagen und Bibliotheken

Dieser Anhang enthält Anleitungen, die Schülern einen schnellen Einstieg in die Arbeit mit POV-Ray ermöglichen sollen. Zum Teil setzen diese Anleitungen die Verwendung von Vorlagedateien (`vorlage.pov`, `anageoL.pov`, `anageoR.pov`, `video.pov`, `video.ini`) voraus (siehe dazu auch die Abschnitte 3.5 und 4.2). Diese Vorlagen sind sehr kurz und rufen vor allem Bibliotheken (Include-Dateien mit der Dateiergung `.inc`) auf, die Definitionen von Kameras, Lichtquellen und Texturen sowie Makros enthalten. Die Makros generieren Koordinatensysteme und einige in der analytischen Geometrie häufig benötigte Objekte, für die POV-Ray selbst keine geeigneten Anweisungen bereitstellt.

Alle Vorlagen und Include-Dateien sowie die in den Anleitungen erwähnten Beispiele befinden sich in der unter <http://www.afiller.de/habil> verfügbaren Materialsammlung zu der vorliegenden Arbeit (in dem Ordner **Vorlagen+Anleitungen**)<sup>1</sup> und können mit POV-Ray oder einem beliebigen Texteditor geöffnet werden.

Von Vorlagen aufgerufene Include-Dateien müssen sich entweder im selben Ordner befinden wie die jeweils zugehörigen Vorlage- bzw. die daraus erstellten Arbeitsdateien oder im Suchpfad von POV-Ray (z. B. im Verzeichnis **POV-Ray for Windows v3.6\include** oder in einem anderen Verzeichnis, das in der Initialisierungsdatei `povray.ini` eingetragen ist) liegen. Da bei Netzwerkinstallationen Benutzer weder Dateien in Programmverzeichnis kopieren noch Initialisierungsdateien ändern können, empfiehlt es sich meist, Vorlage-, Arbeits- und Include-Dateien in denselben Ordner zu kopieren.

Einige der in den Include-Dateien enthaltenen Definitionen und Makros, die für den Einsatz von POV-Ray im Stoffgebiet Analytische Geometrie erstellt wurden, sind am Ende dieses Anhangs abgedruckt. Aus Platzgründen erfolgt hierbei eine Beschränkung auf Auszüge. Die Makros sind nicht für die Bearbeitung durch Schüler vorgesehen; ihr Abdruck dient lediglich der Information über die Art und Weise, wie die in den Anleitungen beschriebenen Objekte generiert werden.

---

<sup>1</sup>Zur Strukturierung und Verwendung der Materialien siehe Anhang D, S. 365f. Die hier beschriebenen Anleitungen und Dateien stehen außerdem auf der u. a. für Schüler und Lehrer gestalteten Internetseite <http://www.afiller.de/3dceg> (Rubrik Downloads) zur Verfügung.

## Kurzanleitung für die Arbeit mit POV-Ray

### Ein erstes Bild mit POV-Ray rendern

1. Laden Sie das Programm von der POV-Ray-Homepage [www.povray.org](http://www.povray.org) herunter und installieren Sie es mit den vorgeschlagenen Standardeinstellungen.
2. Laden Sie die Datei `vorlagen.exe` von der Internetseite [www.afilier.de/3dcdg](http://www.afilier.de/3dcdg) (Rubrik Download) herunter und entpacken Sie diese in einen Arbeitsordner (`vorlagen.exe` ausführen). Die Dateien `vorlage.pov` und `vorlage.inc` sowie das Beispiel `povstart.pov` befinden sich dann im Unterordner `\vorlagen`.<sup>1</sup>
3. Rufen Sie POV-Ray vom Desktop oder über das Windows-Startmenü → Programme → POV-Ray auf. Es öffnet sich das abgebildete Programmfenster.



4. Öffnen Sie eine (möglichst einfache) Datei, z. B. `povstart.pov`. Sie sehen jetzt den Inhalt der Datei (Quelltext) im Bearbeitungsfenster.
5. Wählen Sie (links oben im POV-Ray-Fenster) eine Auflösung für das Bild, das Sie im nächsten Schritt berechnen. Höhere Auflösungen führen zu größeren, detailreicheren Bildern, verlängern aber die Renderzeiten. Die Einstellung AA (Anti-Aliasing) bewirkt eine höhere Bildqualität, verlangsamt aber ebenfalls den Rendervorgang. Wählen Sie für erste Versuche 640x480, No AA oder 640x480, AA 0.3.
6. Starten Sie nun durch einen Klick auf den Button RUN den Rendervorgang. Es öffnet sich ein neues Fenster, in dem das berechnete Bild angezeigt wird.

Sie haben damit Ihr erstes Bild mit POV-Ray gerendert. Sie können Veränderungen daran vornehmen, indem Sie im Bearbeitungsfenster die Größe und Position der Objekte `sphere`, `cone`, `torus` und `cylinder` verändern oder durch Änderung der Kamerakoordinaten (`location <x,y,z>`) von einem anderen Blickpunkt auf die Szene schauen. Die berechneten Bilder befinden sich nach dem Rendern in demselben Ordner wie die POV-Datei. Es handelt sich dabei um Dateien im `.bmp`-Format (mit demselben Namen wie die POV-Datei), die z. B. in einen Text eingefügt oder mit einem Bildbearbeitungsprogramm bearbeitet werden können.

**Hinweis:** Sie müssen bei der Arbeit mit POV-Ray unbedingt *Groß- und Kleinschreibung* beachten und die richtigen *Klammertypen* verwenden. Die Verwendung zusätzlicher Leerzeichen ist aber möglich.

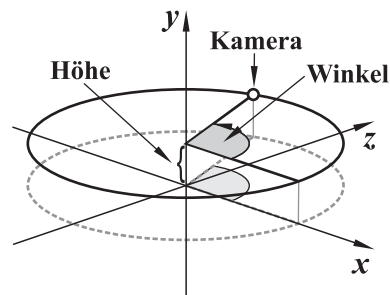
<sup>1</sup>Alternativ lassen sich auch die (identischen) Dateien aus dem Ordner `Vorlagen+Anleitungen` der Materialsammlung zu der vorliegenden Arbeit (siehe S. 365f.) verwenden.



## Erstellung eigener Grafiken unter Verwendung von Vorlagen

Gerade anfangs können die vielfältigen Möglichkeiten von POV-Ray verwirren. Deshalb empfiehlt es sich, eine Datei mit bereits eingefügten Kopfzeilen, Lichtern und definierter Kamera als Vorlage zu verwenden. Dazu können Sie die Datei `vorlage.pov` nutzen.<sup>2</sup> Speichern Sie ihre Datei nach dem Öffnen der Vorlagedatei zunächst unter einem anderen Namen. Fügen Sie geometrische Objekte ein und ordnen Sie diese um den Koordinatenursprung herum an.

Die vorbereitete Kamera ist auf den Koordinatenursprung gerichtet und erfasst (in etwa) diejenigen Punkte, deren sämtliche Koordinaten zwischen `-intervall` und `intervall` liegen; den Wert für `intervall` können Sie verändern. Außerdem lässt sich die Kameraposition durch die Festlegung von `winkel` und `hoehe` ändern.



### Koordinatensystem

Durch Eingabe von `ks` erzeugen Sie ein Koordinatensystem in der Szene, das Ihnen die Orientierung erleichtern kann.

### Geometrische Objekte

**Kugel:** `sphere{<x,y,z>, r}`

(Koordinaten des Mittelpunktes und Radius),

**Quader:** `box{<x1,y1,z1>, <x2,y2,z2>}`

(Koordinaten zweier diagonal gegenüberliegender Eckpunkte),

**Gerader Kreiszylinder:** `cylinder{<x1,y1,z1>, <x2,y2,z2>, r}`

(Koordinaten der Mittelpunkte von Grund- und Deckfläche, Radius),

**Gerader Kreiskegelstumpf:** `cone{<x1,y1,z1>, r1, <x2,y2,z2>, r2}`

(Koordinaten der Mittelpunkte sowie Radien von Grund- und Deckfläche),

**Ebene:** `plane{<A,B,C>, D}`

erzeugt die Ebene mit der Gleichung  $Ax + By + Cz = D$ . Dieser Befehl erzeugt eine „wirkliche“ (unendlich ausgedehnte) Ebene und ist deshalb vor allem für die Erzeugung einer „Grundebene“ (Erdboden) geeignet.

### Texturen

In den Beschreibungen geometrischer Objekte sollte jeweils vor der letzten Klammer eine Oberfläche (Textur) definiert werden, ansonsten werden die Objekte schwarz dargestellt. Obwohl die Erzeugung von Texturen interessant ist, hält sie bei der Erstellung der ersten Szenen auf. Deshalb können Sie zunächst vorgefertigte Texturen verwenden. Folgende sehr einfach zu nutzende Texturen stehen in der Vorlage zur Verfügung:

`mattweiss`, `schwarz`, `blau_matt`, `blau_glanz`, `blau_transp`, `rot_matt`, `rot_glanz`, `rot_transp`, `gruen_matt`, `gruen_glanz`, `gruen_transp`, `schachbrett`, `holz`, `stein`, `silber`, `silbergrau`.

Sie können diese Texturen wie in dem folgenden Beispiel (anstelle von `holz`) beliebigen Objekten zuordnen.

**Beispiel:** `sphere { <0,-1,5> 1 texture {holz} }` erzeugt eine Holzkugel.

<sup>2</sup>In der Datei `vorlage.pov` wird die Datei `vorlage.inc` aufgerufen, in der Kamera, Lichtquellen u. a. definiert sind. Diese Datei muss sich im selben Verzeichnis wie `vorlage.pov` befinden.

## Kurzanleitung: Transformationen, Boolesche Operationen, Prismen und Rotationskörper in POV-Ray

*In dieses Anleitungsmaterial wurden (in modifizierter Form) Beiträge von F. RIEPER einbezogen, der Materialien für ein Unterrichtsprojekt (siehe Abschnitt 5.3) anfertigte.*

### Transformationen

Werden die folgenden Befehle vor der letzten Klammer eines Objekts eingefügt, so wird das entsprechende Objekt verschoben, gedreht sowie gestreckt bzw. gestaucht:

`translate <x,y,z> ,    rotate < $\varphi_x, \varphi_y, \varphi_z$ > ,    scale <x,y,z>.`

#### Beispiele:

`translate <1,2,3>` verschiebt ein Objekt um 1 Einheit in  $x$ -, 2 Einheiten in  $y$ - und 3 Einheiten in  $z$ -Richtung;

`rotate <90,15,30>` dreht ein Objekt  $90^\circ$  um die  $x$ -Achse,  $15^\circ$  um die  $y$ -Achse und  $30^\circ$  um die  $z$ -Achse;

`scale <2,3,0.5>` vergrößert ein Objekt auf die zweifache  $x$ - und die dreifache  $y$ -Ausdehnung und staucht es auf die halbe  $z$ -Ausdehnung zusammen.

Diese Transformationen können Sie in beliebiger Kombination und Reihenfolge in die Beschreibungen von Objekten einfügen. Beachten Sie aber, dass das Ergebnis bei der Nacheinanderausführung mehrerer Transformationen von der Reihenfolge abhängt.

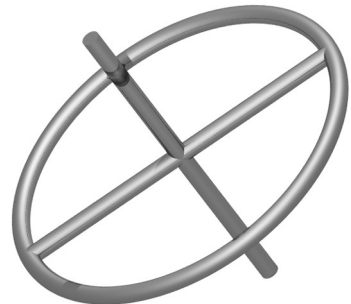
### Boolesche Operationen: Vereinigungen, Schnitte, Differenzen

#### Vereinigung: `union`

Durch Vereinigungen können komplexere Gebilde erzeugt werden. Transformationen und Texturen müssen dann nicht für jedes Teilobjekt beschrieben werden, sondern nur einmal für das gesamte Vereinigungsobjekt.

#### Beispiel:<sup>1</sup>

```
union { cylinder {<-2,0,0>, <2,0,0>, 0.1}
      cylinder {<0,-2,0>, <0,2,0>, 0.1}
      torus {2,0.1}
      rotate <-40,-30,40>
      texture { silbergrau } }
```



#### Erläuterungen:

<code>union { ... }</code>	Die Vereinigung enthält beliebige Objekte, die dann als ein Ganzes behandelt werden. Die Eigenschaften des Gesamtobjekts werden am Ende – nach der Definition der Teilobjekte – angegeben.
<code>rotate &lt;-40,-30,40&gt;</code>	Die Drehung bezieht sich auf alle Teilobjekte, die zur Vereinigungsmenge gehören.
<code>texture { ... }</code>	Auch die Textur wird allen Teilobjekten zugewiesen.

<sup>1</sup>Alle in dieser Anleitung gegebenen Beispiele können durch Einsetzen der Anweisungen in die Vorlagdatei `vorlage.pov` direkt gerendert werden.

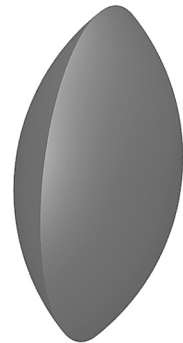
**Schnittmenge: intersection**

Wie in der Mengenlehre lassen sich mit POV-Ray Schnittmengen bilden. So entsteht aus zwei gleich großen Kugeln, die sich schneiden, eine Linse:

```
intersection { sphere {<-1,0,0>,1.5}
                sphere {<1,0,0>,1.5}
                rotate <0,-10,0>
                texture { silbergrau } }
```

**Erläuterung:** Durch den Befehl `intersection` entsteht der Durchschnitt der Teilobjekte, für zwei versetzte Kugeln also eine Linse.

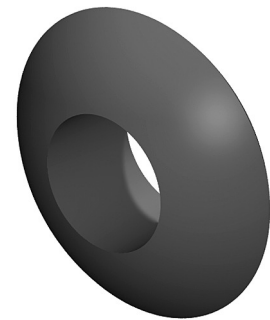
**Aufgabe:** Bilden Sie den Durchschnitt aus einem Würfel und einer Kugel derart, dass ein „Würfel mit abgerundeten Ecken“ entsteht.

**Differenz (heraus schneiden): difference**

```
difference { intersection { sphere {<-1,0,0>,1.5}
                          sphere {<1,0,0>,1.5} }
            cylinder {<-3,0,0>, <3,0,0>, 0.5}
            rotate <0,-78,30>
            translate <0.2,0.1,0>
            texture { blau_matt } }
```

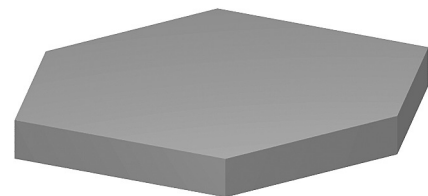
**Erläuterung:** Der Befehl `difference` schneidet aus der Linse (wie oben als Durchschnittsmenge zweier Kugeln erzeugt) einen Zylinder heraus. Die *Reihenfolge*, in der die Objekte genannt werden, ist hierbei wichtig.

**Aufgabe:** Schneiden Sie aus einem Würfel eine Kugel so heraus, dass ein Hohlwürfel entsteht.

**Prismen und Pyramiden**

Durch Festlegung der Eckpunktkoordinaten der Grundfläche in der  $x$ - $z$ -Ebene lassen sich in POV-Ray Prismen und Pyramiden erzeugen.

```
prism { linear_sweep linear_spline
        0, // Höhe der Grundfläche
        1, // Höhe der Deckfläche
        7, // Zahl der Eckpunkte + 1
        <3,5>, <-3,5>, <-5,0>, <-3,-5>,
        <3, -5>, <5,0>, <3,5>
        texture { silbergrau } }
```

**Erläuterungen:**

`linear_sweep` Festlegung eines geraden Prismas; `conic_sweep` erzeugt eine Pyramide.

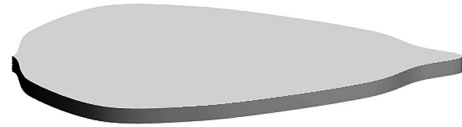
`linear_spline` Legt fest, dass die Grundfläche des Prismas geradlinig begrenzt ist. Die Grundfläche kann auch von Kurven (Splines) begrenzt sein (siehe unten).

`<3,5> ... <3,5>` Die Koordinatenpaare legen die Grundfläche des Prismas in der  $x$ - $z$ -Ebene fest. Anfangs- und Endpunkt müssen übereinstimmen. Die Zahl der anzugebenden Punkte ist daher um 1 höher als die der tatsächlichen Eckpunkte.

**Hinweis:** Zur Änderung der Lage eines Prismas müssen die Befehle `rotate` und `translate` verwendet werden.

## Prismen mit Grundflächen, die von gekrümmten Kurven begrenzt sind

Die Erstellung eines Prismas mit einer Grundfläche, die durch eine „glatte“ Kurve (kubische Splinekurve) begrenzt wird, ist oft interessanter als die Erzeugung normaler Prismen.



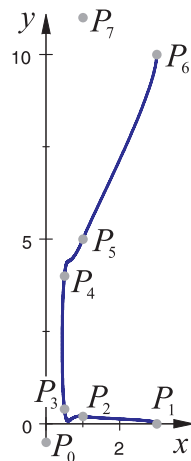
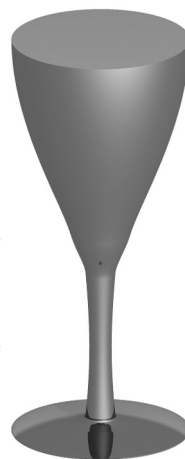
```
prism { cubic_spline, 0, 0.2          // Höhen der Grund- und Deckfläche
        17,                          // Gesamtzahl der Punkte
        <-2.5,0>                      // Kontrollpunkt
        <-2,0>, <-1.8,-0.5>, <-1,-1>, <0,-1>, // Punkte, durch welche die
        <1,-0.8>, <2,-0.5>, <2.5,-0.2>, <3,0>, // Kurve verläuft; der erste
        <2.5,0.2>, <2,0.5>, <1,0.8>, <0,1>, // und letzte Punkt müssen
        <-1,1>, <-1.8,0.5>, <-2,0>          // identisch sein
        <-2.5,0>                          // Kontrollpunkt
        texture{gruen_matt}                }
```

Die beiden Kontrollpunkte bestimmen die Richtungen, in welche die Begrenzungskurve der Grundfläche im ersten ihrer Punkte (der mit dem letzten Punkt übereinstimmt) weist.

## Rotationskörper

Durch Körper, die von Rotationsflächen (**sor** – Surface of Revolution) begrenzt werden, lassen sich viele reale Gegenstände modellieren. Zur Erzeugung eines Rotationskörpers müssen Koordinaten von Punkten eingegeben werden, durch die POV-Ray eine glatte Kurve (kubische Splinekurve) legt und diese um die  $y$ -Achse rotiert.

```
sor { 8, // Zahl der Stütz- + Kontrollpunkte
      <0,-0.5>, // Kontrollpunkt
      <3,0>, <1,0.2>, <0.5,0.4>, // Punkte der er-
      <0.5,4>, <1,5>, <3,10>, // zeugenden Kurve
      <1,11> // Kontrollpunkt
      scale <1,1.3,1>
      texture { silbergrau } }
```



### Erläuterungen und Hinweise:

- POV-Ray verwendet standardmäßig die  $y$ -Achse als Rotationsachse. Die 2D-Koordinaten der Punkte  $P_0$  bis  $P_n$  legen die erzeugende Kurve in der  $x$ - $y$ -Ebene fest.
- Der erste und der letzte Punkt (in der Skizze  $P_0$  und  $P_7$ ) sind Kontrollpunkte, die nicht auf der erzeugenden Kurve liegen, aber Einfluss auf den Kurvenverlauf (Richtung der Tangenten) in  $P_1$  bzw.  $P_6$  haben. Wird ein Rotationskörper mit `sor { 8, ... }` erzeugt, so müssen 6 Kurvenpunkte und 2 Kontrollpunkte festgelegt werden.
- `scale <1,1.3,1>` streckt den Rotationskörper in  $y$ -Richtung um den Faktor 1,3.
- Um die Lage eines Rotationskörpers zu verändern, müssen die Befehle `rotate` und `translate` verwendet werden.
- Der `sor`-Befehl erzeugt „massive“ Rotationskörper. Um Rotationsflächen zu erstellen, können Sie den `lathe`-Befehl verwenden (siehe POV-Ray-Hilfe, Stichwort „Lathe“).

## Darstellung von Punkten, Vektoren, Strecken, Geraden und Ebenen mithilfe von POV-Ray

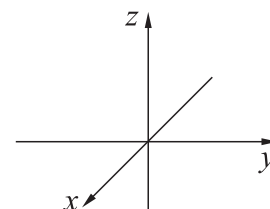
Mithilfe der Erweiterungen `anageoL.inc` bzw. `anageoR.inc` können Sie in POV-Ray Objekte und Lagebeziehungen, die in der analytischen Geometrie behandelt werden, veranschaulichen.<sup>1</sup>

### Hinweis:

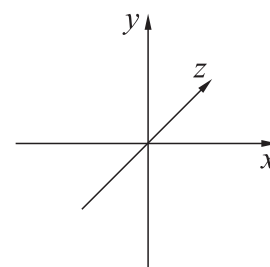
Leider verwenden Computergrafikprogramme wie POV-Ray ein anderes Koordinatensystem (KS) als die meisten Lehrbücher: Während hier ein so genanntes *rechtshändiges KS* gebräuchlich ist, benutzt POV-Ray ein *linkshändiges KS*. Sinnvoll ist es, sich an das KS, das die Software verwendet, zu gewöhnen. Allerdings kann man POV-Ray für selbst erzeugte Befehle das in den Lehrbüchern übliche System „aufzwingen“ – in Wirklichkeit werden dabei alle Koordinaten, die Sie eingeben, im Hintergrund manipuliert.

Sie können sich zwischen zwei Versionen entscheiden:

- `anageoL.inc` (mit der zugehörigen Vorlage `anageoL.pov`) verwendet dasselbe linkshändige KS wie POV-Ray und kann mit allen POV-Ray-Befehlen problemlos ergänzt werden.
- `anageoR.inc` (mit der Vorlage `anageoR.pov`) setzt die in dieser Anleitung beschriebenen Objekte der analytischen Geometrie in das aus Lehrbüchern bekannte rechtshändige KS um. Diese Objekte können dann aber mit den Objekten, die POV-Ray selbst beschreibt, nicht sinnvoll kombiniert werden. Sie sollten diese Version deshalb nur benutzen, wenn Sie mit dem rechtshändigen System vertraut sind und sich nicht umstellen wollen.



Rechtshändiges Koordinatensystem



Linkshändiges Koordinatensystem

Die folgenden Beschreibungen gelten für beide Versionen `anageoL.inc` und `anageoR.inc`.

### Die Befehle des Makropaketes `anageoL.inc` bzw. `anageoR.inc`

`punkt (<x,y,z>, textur)`

Punkt (als kleine Kugel dargestellt) mit den Koordinaten  $(x; y; z)$

`pluspunkt (<x,y,z>, textur)`

Punkt mit Lot auf die  $x$ - $y$ -Ebene und Verbindungsstrecken zu den Koordinatenachsen

`ortsvektor (<x,y,z>, textur)`

Ortsvektor des Punktes  $P(x; y; z)$

`verbindungsvektor (<x1,y1,z1>, <x2,y2,z2>, textur)`

Verbindungsvektor der Punkte  $P_1(x_1; y_1; z_1)$  und  $P_2(x_2; y_2; z_2)$

`vektoranpunkt (<xP,yP,zP>, <x,y,z>, textur)`

Pfeildarstellung des Vektors  $\begin{pmatrix} x \\ y \\ z \end{pmatrix}$  an den Punkt  $P(x_P; y_P; z_P)$  antragen

<sup>1</sup>Diese Dateien und die zugehörigen Vorlagen `anageoL.pov`, `anageoR.pov` befinden sich in dem Ordner `Vorlagen+Anleitungen` (s. S. 365) und stehen auch unter [www.afiller.de/3dcdg](http://www.afiller.de/3dcdg) zur Verfügung.

**strecke** ( $\langle x_1, y_1, z_1 \rangle$ ,  $\langle x_2, y_2, z_2 \rangle$ , *textur*)

Strecke mit den Endpunkten  $P_1(x_1; y_1; z_1)$  und  $P_2(x_2; y_2; z_2)$

**gerade** ( $\langle x_1, y_1, z_1 \rangle$ ,  $\langle x_2, y_2, z_2 \rangle$ , *textur*)

Gerade durch die Punkte  $P_1(x_1; y_1; z_1)$  und  $P_2(x_2; y_2; z_2)$

**ebenepar** ( $\langle x_P, y_P, z_P \rangle$ ,  $\langle x_a, y_a, z_a \rangle$ ,  $\langle x_b, y_b, z_b \rangle$ , *textur*)

Ebene durch den Punkt  $P(x_P; y_P; z_P)$  mit den Richtungsvektoren  $\vec{a} = \begin{pmatrix} x_a \\ y_a \\ z_a \end{pmatrix}$ ,  $\vec{b} = \begin{pmatrix} x_b \\ y_b \\ z_b \end{pmatrix}$

**ebene** (A, B, C, D, *textur*)

Ebene mit der Gleichung  $Ax + By + Cz + D = 0$

**Hinweis:** Beachten Sie bei diesen Befehlen, dass Sie tatsächlich *runde Klammern* (anstatt der ansonsten in POV-Ray zu verwendenden geschweiften Klammern) benutzen. Dieser Unterschied rührt daher, dass die beschriebenen Befehle Makros aufrufen.

**Texturen:** Für *textur* müssen Sie in den Befehlen Texturen angeben.

Beispiel: **ortsvektor** ( $\langle 3, -2, 1 \rangle$ , *silbergrau*)

stellt den Ortsvektor des Punktes  $P(3; -2; 1)$  als silbergrauen Pfeil dar.

Sie können folgende Texturen besonders einfach verwenden:

*mattweiss*, *schwarz*, *blau\_matt*, *blau\_glanz*, *blau\_transp*, *rot\_matt*, *rot\_glanz*, *rot\_transp*, *gruen\_matt*, *gruen\_glanz*, *gruen\_transp*, *holz*, *silber*, *silbergrau*, *schachbrett*

Sinnvoll ist es, für Ebenen teilweise transparente Texturen (*...\_transp*) zu verwenden, da es oft notwendig ist, zu sehen, was hinter den Ebenen liegt.

**Koordinatensystem:** Zur Veranschaulichung der Lage von Objekten können Sie mittels *ks* ein Koordinatensystem einfügen, das sich automatisch der Größe des gewählten Ausschnittes (*intervall*) anpasst.

### Verwenden der Vorlagen *anageoL.pov* bzw. *anageoR.pov*

Kopieren Sie die Dateien *anageoL.inc* und *anageoL.pov* bzw. *anageoR.inc* und *anageoR.pov* gemeinsam in einen Arbeitsordner auf Ihrem PC. Sie können nun mit der Datei *anageoL.pov* bzw. *anageoR.pov* arbeiten und die oben beschriebenen Befehle verwenden. Die POV-Ray-Dateien (Endung *.pov*) müssen sich immer in demselben Ordner befinden wie die zugehörigen *.inc*-Dateien, da sie diese aufrufen.

Die vorbereitete Kamera erfasst (in etwa) diejenigen Punkte, deren sämtliche Koordinaten zwischen *-intervall* und *intervall* liegen; Sie können diesen Wert in der Zeile

```
#declare intervall = 5;
```

verändern. Die Kameraposition können Sie durch Festlegung von *winkel* und *hoehe* ändern.

**Aufgabe** (Ausprobieren einiger Befehle von *anageoL.inc* bzw. *anageoR.inc*):

Öffnen Sie die Datei *anageoL.pov* bzw. *anageoR.pov* und speichern Sie diese zunächst unter einem anderen Namen. Stellen Sie die Punkte  $P(-2; 5; 3)$  und  $Q(2; -1; 4)$ , ihren Verbindungsvektor sowie die Gerade durch diese beiden Punkte dar. Fügen Sie ein Koordinatenkreuz hinzu. Betrachten Sie die Darstellung aus verschiedenen Richtungen, indem Sie die Werte für *winkel* und *hoehe* ändern. Experimentieren Sie mit *punkt* und *pluspunkt*; fügen Sie auch Ortsvektoren hinzu.

## Darstellung von Flächen, die durch Gleichungen oder Parameterdarstellungen gegeben sind, in POV-Ray

### Flächendarstellung durch implizite Gleichungen: `isosurface`

Durch Gleichungen der Form  $f(x, y, z) = 0$  gegebene Flächen werden in POV-Ray durch folgende Anweisungen dargestellt:

<code>isosurface {</code>	
<code>function { f(x,y,z) }</code>	Funktionsterm $f(x, y, z)$ (linke Seite der impliziten Gleichung $f(x, y, z) = 0$ der Fläche)
<code>contained_by { box{a,b} }</code>	dargestellter Ausschnitt der Fläche ( $x$ , $y$ und $z$ jeweils zwischen $a$ und $b$ ); es können auch ein beliebiger Quader <code>box{&lt;a<sub>x</sub>,a<sub>y</sub>,a<sub>z</sub>&gt;,&lt;b<sub>x</sub>,b<sub>y</sub>,b<sub>z</sub>&gt;}</code> oder eine Kugel angegeben werden, um den dargestellten Ausschnitt zu begrenzen
<code>accuracy 0.001</code>	Genauigkeit der Berechnung der Fläche (kleinere Werte führen zu einer genaueren und langsameren Darstellung)
<code>max_gradient 4</code>	maximaler Anstieg (große Werte → langsame Berechnung)
<code>open</code>	die Ränder bleiben offen; ohne diesen Befehl werden die Seitenflächen der Containerbox mit dargestellt
<code>}</code>	

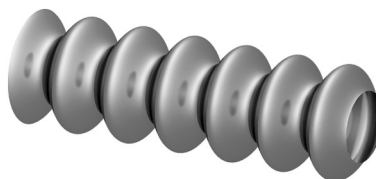
Der Wert `max_gradient` beschreibt den maximalen Anstieg der Funktion  $f(x, y, z)$ ; dieser errechnet sich aus den Ableitungen der Funktion nach  $x$ ,  $y$  und  $z$ . Setzt man `max_gradient` zu klein, so werden nicht alle Bereiche der Fläche korrekt dargestellt; große Werte führen zu langen Renderzeiten. Um optimale Werte für `max_gradient` und `accuracy` zu finden, ist etwas Experimentieren notwendig. Die angegebenen Werte haben sich als sinnvolle Ausgangswerte für viele Anwendungen erwiesen.

**Beispiel 1:** Die Fläche mit der impliziten Gleichung

$$\frac{\sin(10x)}{8} + \sqrt{z^2 + y^2} - 0,5 = 0$$

kann durch die Eingabe folgender Zeilen in POV-Ray dargestellt werden:<sup>1</sup>

```
isosurface {
    function { sin(10*x)/8 + sqrt(z*z+y*y) - 0.5 }
    contained_by { box { -2, 2 } }
    accuracy 0.005    max_gradient 10
    open    texture{ silbergrau }
}
```



**Beispiel 2:** Durch die Gleichung

$$-3x^2 + 2y^2 - 3z^2 - 0,25 = 0$$

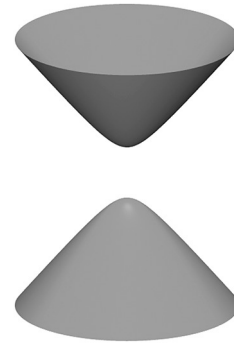
wird ein zweischaliges Hyperboloid<sup>2</sup> beschrieben.

<sup>1</sup>Alle in dieser Anleitung gegebenen Beispiele können durch Einsetzen der Anweisungen in die Vorlagedatei `vorlage.pov` direkt gerendert werden.

<sup>2</sup>Hyperboloide sind Flächen 2. Ordnung (Quadriken). Für diese Flächen gibt es in POV-Ray auch den speziellen Befehl `quadric` (siehe POV-Ray-Hilfe), der meist schneller zu denselben Ergebnissen führt, aber nicht so universell ist wie `isosurface`.

In POV-Ray entsteht es durch die folgende Eingabe:

```
isosurface {
  function { - 3*x*x + 2*y*y - 3*z*z - 0.25 }
  contained_by { box { -1.8, 1.8 } }
  accuracy 0.01
  max_gradient 20
  open
  texture { rot_matt }
  no_shadow
}
```



## Flächendarstellung durch Parametergleichungen: **parametric**

Durch Parametergleichungen gegebene Flächen stellt POV-Ray durch folgende Eingabe dar:

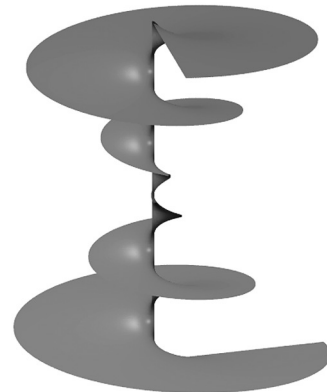
<b>parametric {</b>	
function { $x(u,v)$ }	Funktionen, durch welche die Fläche gegeben ist:
function { $y(u,v)$ }	Koordinaten $x$ , $y$ und $z$ in Abhängigkeit von den
function { $z(u,v)$ }	Parametern $u$ und $v$
$\langle u_1, v_1 \rangle, \langle u_2, v_2 \rangle$	Definitionsbereiche für die Parameter $u$ und $v$
contained_by { box{a,b} }	Begrenzungsquader (siehe <b>isosurface</b> )
accuracy 0.001    max_gradient 4	Genauigkeit der Berechnung, maximaler Anstieg
precompute 18 x,y,z	} Vorausberechnung von Intervallen (siehe unten)

**Beispiel:** Die konische Spirale mit der Parameterdarstellung

$$\begin{aligned} x &= u \cdot v \cdot \sin(15v) \\ y &= v \\ z &= u \cdot v \cdot \cos(15v) \end{aligned} \quad (\text{mit } u \in [0; 1], v \in [-1; 1])$$

lässt sich durch die folgende Eingabe darstellen:

```
parametric {
  function {u*v*sin(15*v)}
  function {v}
  function {u*v*cos(15*v)}
  <0,-1>,<1,1>
  contained_by { box { -1, 1 } }
  max_gradient 10    accuracy 0.0005
  precompute 18 x,y,z
  texture{blau_matt}    no_shadow    }
```



Die Anweisung **precompute 18 x,y,z** bewirkt eine Unterteilung der Fläche in  $2^{18}$  Teilflächen. Für jede dieser Teilflächen werden die Intervalle vorausberechnet, in denen die Koordinaten  $x$ ,  $y$  und  $z$  liegen. Dies erfordert eine gewisse Vorausberechnungszeit (parsing) vor dem Rendervorgang, verkürzt diesen dann jedoch erheblich.

Geeignete Werte für **accuracy** und **max\_gradient** müssen wie bei dem Befehl **isosurface** durch Versuche gefunden werden. Bei zu geringen Werten für **max\_gradient** werden u. U. nicht alle Bereiche der Fläche dargestellt, große Werte verlängern erheblich den Rendervorgang. Flächen werden korrekt berechnet, wenn der angegebene Wert für **max\_gradient** mindestens so groß ist wie das Maximum aller 6 partiellen Ableitungen der Koordinatenfunktionen nach den Parametern  $u$  und  $v$ , also  $\frac{\partial x}{\partial u}$ ,  $\frac{\partial x}{\partial v}$ ,  $\frac{\partial y}{\partial u}$ ,  $\frac{\partial y}{\partial v}$ ,  $\frac{\partial z}{\partial u}$  und  $\frac{\partial z}{\partial v}$ , über dem gesamten Definitionsbereich von  $u$  und  $v$ .



## Kurzanleitung: Mit POV-Ray Videos erstellen

Während für die Erzeugung eines Bildes nur eine Datei (die POV-Ray-Datei mit der Endung `.pov`) erforderlich ist, wird für die Erzeugung von Videos zusätzlich eine Initialisierungsdatei (Endung `.ini`) mit Steuerinformationen für die Erzeugung des Videos benötigt. Die Initialisierungsdatei enthält vor allem die Information, wie viele Einzelbilder (Frames) erzeugt werden sollen und wie diesen Einzelbildern Werte für den Parameter `clock` (gewissermaßen die Uhr) zugeordnet werden. Weiterhin enthält sie den Aufruf der POV-Ray Datei, welche die zu berechnende Szene beschreibt. Eine vollständige `.ini`-Datei ist folgendermaßen aufgebaut:<sup>1</sup>

```
Antialias=On           ; Festlegungen zur Qualität
Antialias_Threshold=0.2 ; der berechneten Bilder
Antialias_Depth=3      ; (Antialiasing)
Input_File_Name=video.pov ; Name der POV-Ray-Datei, welche die Szene enthält
Initial_Frame=1        ; Nummer des ersten Bildes (Frame)
Final_Frame=36         ; Nummer des letzten Bildes (Frame)
Initial_Clock=0         ; Werte für den clock-Parameter, die dem
Final_Clock=1          ; ersten bzw. letzten Bild entsprechen
Pause_when_Done=off    ; Die Einzelbilder werden ohne Pausen hintereinander
Cyclic_Animation=off   ; erzeugt; die Animation wird nur 1 mal gerendert.
```

Wichtig sind dabei vor allem die folgenden Einstellungen, alle anderen Zeilen können Sie unverändert lassen:

<code>Input_File_Name=video.pov</code>	Hier muss der Dateiname der POV-Ray-Datei angegeben werden, die als Vorlage für das Video dient und in welcher der <code>clock</code> -Parameter verwendet wurde.
<code>Initial_Frame=1</code> <code>Final_Frame=36</code>	Diese beiden Werte bewirken, dass insgesamt 36 Bilder erzeugt werden.
<code>Initial_Clock=0</code> <code>Final_Clock=1</code>	Die Werte für den Parameter <code>clock</code> erhöhen sich vom ersten bis zum letzten Bild in gleichen Abständen je Bild. Den Anfangs- und den Endwert können Sie hier festlegen.

**Hinweis:** `Clock` muss in `.ini`-Dateien mit großem `C`, in POV-Ray-Dateien jedoch mit kleinem `c` (also `clock`) geschrieben werden.

Damit ein Video entsteht, müssen Objekte der Szene von dem Parameter `clock` abhängig sein – dieser Parameter verändert sich zeitabhängig und steuert das Video. In dem Beispielfile `video.pov` hängt nur die Position der Kamera von `clock` ab – die Kamera bewegt sich auf einer Kreisbahn. Um mit der technischen Vorgehensweise der Erstellung eines Videos vertraut zu werden, können Sie die folgenden Schritte nachvollziehen:

1. Kopieren Sie die Dateien `video.ini` und `video.pov` in einen Arbeitsordner auf Ihrer Festplatte.<sup>2</sup>
2. Öffnen Sie die Initialisierungsdatei `video.ini` in POV-Ray, ändern Sie evtl. den Wert für `Final_Frame` (in Abhängigkeit davon, wie viele Einzelbilder Sie berechnen möchten).

<sup>1</sup>Sie müssen diese Zeilen nicht abschreiben, sondern können die Datei `video.ini` verwenden.

<sup>2</sup>Die Dateien `video.ini` und `video.pov` sowie die hier noch benötigte Software `pjbm2avi` befinden sich in dem Ordner `Vorlagen+Anleitungen` (siehe S. 365f.) und sind auch in der Datei `vorlagen.exe` enthalten, die unter [www.afiller.de/3dgc](http://www.afiller.de/3dgc) (Rubrik Download) heruntergeladen werden kann.

3. Rendern Sie diese Initialisierungsdatei (Klick auf RUN-Button in POV-Ray). Der Rendervorgang nimmt einige Zeit in Anspruch, da (bei Beibehaltung der Voreinstellung für `Final_Frame`) 36 Einzelbilder berechnet werden.
4. Nach Abschluss des Renderns befinden sich 36 Bilddateien `video01.bmp ... video36.bmp` in demselben Ordner wie die Dateien `video.ini` und `video.pov`.
5. Um aus den 36 Einzelbildern ein Video im `.avi`-Format zu erzeugen, können Sie das frei verfügbare Programm `pjbmp2avi` verwenden (Anleitung siehe unten).

Sie können die beiden Dateien `video.ini` und `video.pov` variieren, um eigene Animationen zu erstellen. Es lassen sich beliebige Koordinaten, Transformationen und andere Werte, die in Szenen auftreten, animieren – also zeitabhängig verändern. Dazu müssen sie als Funktionen des Parameters `clock` ausgedrückt werden. Um Bewegungsabläufe zu erstellen, müssen die Koordinaten von Punkten parameterabhängig angegeben werden. Es sind dazu also Kurven des Raumes (die Bewegungsbahnen) durch Parameterdarstellungen zu beschreiben.

### Erzeugen einer Videodatei mit pjbmp2Avi

Löschen oder verschieben Sie zunächst aus dem Verzeichnis, in dem sich die Bilddateien für das Video befinden, alle `.bmp`-Bilddateien, die nicht für das Video genutzt werden sollen. Starten Sie dann das Programm `pjbmp2Avi` („Bitmap to Avi“). Es erscheint das abgebildete Fenster. Nehmen Sie die folgenden Einstellungen vor.

**Directory:** Wählen Sie den Ordner aus, in dem die Bilder liegen – dies ist der Ordner der POV-Ray-Datei, falls Sie die Bilder nicht verschoben haben.

**Input name:** Geben Sie die ersten Buchstaben der Dateinamen der Bilddateien ein; die Datei-Liste (**Files**) füllt sich dann von selbst.

**Files:** Prüfen Sie, ob wirklich alle Bilder durchgehend nummeriert sind. Andere Bilder müssen ggf. aus dem Verzeichnis entfernt werden (siehe oben).

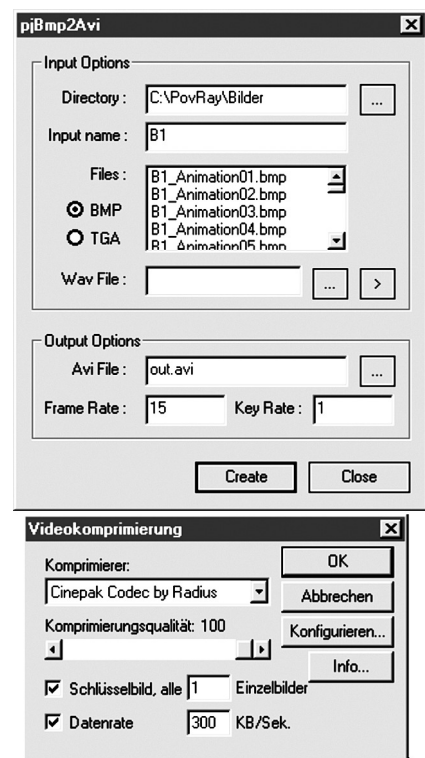
**Wav File:** Diese Einstellung wird nur benötigt, wenn Sie Musik hinter ihr Video legen wollen.

**Avi File:** Geben Sie ihrem Film einen Namen und wählen Sie das Verzeichnis, in dem er gespeichert werden soll. Standardmäßig heißt er `out.avi`.

**Frame Rate:** Dieser Wert gibt an, wie viele Bilder in einer Sekunde abgespielt werden. Diesen Wert müssen Sie nicht verändern, auch nicht **Key Rate**.

Klicken Sie im Anschluss auf **Create** und es erscheint das Fenster **Videokomprimierung**. Wählen Sie bei **Komprimierer** z. B. „Indeo Video“ oder „Cinepak“.

Die vorgeschlagenen Einstellungen für die Komprimierungsqualität, die Schlüsselbilder und die Datenrate sind meist sinnvoll. Nach einem Klick auf **OK** wird das Video erzeugt. Nach der Umwandlung können Sie das `.avi`-Video z. B. mit dem Windows Media Player betrachten.



## Vorbereitete Objekte für den Einstieg in POV-Ray

Im Folgenden sind Teile der Include-Datei `vorlage.inc` abgedruckt, die Schülern (im Zusammenhang mit der Vorlagedatei `vorlage.pov`) zur Verfügung gestellt werden kann, um schnell erste Szenen in POV-Ray selbst zu erstellen (siehe Abschnitt 3.5, S. 212). Dazu enthält die Include-Datei Festlegungen für die Kamera und zwei Lichtquellen, vorgefertigte Texturen, die mit einfachen Namen aufgerufen werden können, sowie ein Makro für die Darstellung eines Koordinatensystems mit Achsenbezeichnungen und Skaleneinteilung. Dessen Größe passt sich automatisch dem darzustellenden Teil des Raumes (beschrieben durch die Variable `intervall`) an. Die Schüler müssen die hier wiedergegebene Datei nicht öffnen; die in ihr enthaltenen Definitionen werden aufgerufen, wenn sie mit der Datei `vorlage.pov` arbeiten und die in der Anleitung (siehe S. 330f.) genannten Texturen verwenden bzw. den Befehl `ks` aufrufen.

```
// vorlage.inc: Festlegung der Kamera, Lichtquellen, Texturen, Koord.-system
// -----
// Beschreibung der Kamera
// -----
camera { orthographic
    location <60*intervall*cos(winkel*pi/180),
              5*hoehe*intervall,
              60*intervall*sin(winkel*pi/180)>
    right x*4/3*2.7*intervall up y*2.7*intervall
    look_at <0, -intervall/10, 0> }

// -----
// Lichtquellen
// -----
light_source { <2.5*intervall,2*intervall,-5*intervall> color rgb <1,1,1> }
light_source { <0,10*intervall,2*intervall> color rgb <1,1,1> }
// -----
// Definition von Materialien und Oberflächen (Texturen)
// -----
#include "colors.inc"
#include "textures.inc"
#include "stones1.inc"
#include "metals.inc"
#declare holz = DMFWood6;
#declare stein = T_Stone9;
#declare silber = T_Silver_4D;
#declare blau_glanz = texture { pigment { color rgb <0.1,0.1,1.0> }
                                finish { ambient 0.05 diffuse 0.3 reflection 0.4
                                           phong 0.4 phong_size 50 }
                                };

    In analoger Weise werden an dieser Stelle weitere Texturen definiert.
#declare schachbrett = texture {
    pigment { checker color rgb <0.7,0.7,1>, color rgb <0.2,0.2,0.2> }
    finish { ambient 1.0 brilliance 2 diffuse 0.7 metallic
              specular 0.9 roughness 1/20 phong 0.45 reflection 0 }
    scale intervall*0.5
};
```

```

// Makro für die Darstellung eines Koordinatensystems
// -----
#macro koordinatensystem ( achsenlaenge )
#declare kstextur = texture { pigment { rgb<0.1,0.1,0.3> }
                                finish { ambient 0.0 diffuse 0.1 reflection 0.1
                                           brilliance 1 specular 0.3 roughness 0.2 } };
#declare KSFont="cyrvetic.ttf" ;
#declare KSSchriftgroesse=achsenlaenge/5 ;
#declare skalenkugelradius=achsenlaenge/70;
union{
// x-Achse:
cylinder{-x*1.1*achsenlaenge , x*1.05*achsenlaenge, achsenlaenge/120}
cone{ x*1.2*achsenlaenge, 0 , x*1.05*achsenlaenge, achsenlaenge/48}
#declare Count=-achsenlaenge;
#while (Count < achsenlaenge+1)
    sphere {<Count,0,0>, skalenkugelradius}
#declare Count=Count+1;
#end
// y-Achse:
cylinder{-y*1.1*achsenlaenge , y*1.05*achsenlaenge, achsenlaenge/120}
cone{ y*1.2*achsenlaenge, 0 , y*1.05*achsenlaenge, achsenlaenge/48}
#declare Count=-achsenlaenge;
#while (Count < achsenlaenge+1)
    sphere {<0,Count,0>, skalenkugelradius}
#declare Count=Count+1;
#end
// z-Achse:
cylinder{-z*1.1*achsenlaenge , z*1.05*achsenlaenge, achsenlaenge/120}
cone{ z*1.2*achsenlaenge, 0 , z*1.05*achsenlaenge, achsenlaenge/48}
#declare Count=-achsenlaenge;
#while (Count < achsenlaenge+1)
    sphere {<0,0,Count>, skalenkugelradius}
#declare Count=Count+1;
#end
// Achsenbezeichnungen:
text{ttf KSFont "x",0.1,0 scale KSSchriftgroesse
      translate <1.08*achsenlaenge,-achsenlaenge/7,0>}
text{ttf KSFont "y",0.1,0 scale KSSchriftgroesse
      translate <-achsenlaenge/8,1.09*achsenlaenge,0>}
text{ttf KSFont "z",0.1,0 scale KSSchriftgroesse
      translate <achsenlaenge/20,0,1.1*achsenlaenge> }
texture {kstextur}    no_shadow
#end
// Vereinfachter Aufruf des Koordinatensystems (ks)
#declare ks = koordinatensystem (int(intervall))

```

## Makros für die Darstellung von Punkten, Strecken, Vektoren (als Pfeile), Geraden und Ebenen

Im Folgenden sind Teile der Include-Datei `anageoR.inc` wiedergegeben, die Makros für die Darstellung der im gegenwärtigen Unterricht der analytischen Geometrie hauptsächlich behandelten Objekte enthält: Punkte (dargestellt als kleine Kugeln), Vektoren (als Pfeile), Strecken, Geraden und Ebenen (in Parameter- und Koordinatenform). Diese Objekte werden in den Makros mithilfe einfacher Grundkörper (Kugeln, Zylinder, Kegel und Quader) konstruiert und in Abhängigkeit von den Koordinateneingaben des Benutzers positioniert. Auf die Verwendung der Befehle `isosurface` und `parametric` wurde verzichtet, da die Berechnung von Ebenen bei Verwendung dieser Befehle ein Vielfaches der Zeit benötigen würde, die für die Erzeugung und geeignete Positionierung dünner Quader notwendig ist.

Die Dateien `anageoR.inc` bzw. `anageoL.inc` enthalten weiterhin Beschreibungen einer Kamera, von Lichtquellen und Texturen sowie eines Koordinatensystems; diese entsprechen den bereits in der Datei `vorlage.inc` verwendeten Definitionen und Makros (siehe S. 341). Die Schüler müssen die Dateien `anageoR.inc` bzw. `anageoL.inc` nicht öffnen; die in ihnen enthaltenen Makros werden aufgerufen, wenn sie die entsprechenden Vorlagen `anageoR.pov` bzw. `anageoL.pov` nutzen (wie in der Anleitung auf S. 335f. beschrieben). Die Makros werden hier nur zu Informations- und Dokumentationszwecken wiedergegeben.

Wie schon aus der Anleitung auf S. 335f. hervorgeht, liegen die Vorlagen und Makros in zwei Varianten vor. Die `anageoL`-Version stellt alle Objekte in dem von POV-Ray verwendeten linkshändigen Koordinatensystem dar, während bei der `anageoR`-Variante eine Transformation in ein rechtshändiges Koordinatensystem mit der in Schulbüchern meist aufzufindenden Achsenausrichtung erfolgt. Dazu werden Benutzereingaben  $(x, y, z)$  die Tripel  $(y, z, -x)$  zugeordnet bzw. es wird auf anhand von POV-Ray-Koordinaten generierte Objekte die Transformationsmatrix

angewendet. Die `anageoR`-Version soll die Nutzung von POV-Ray

$$\begin{pmatrix} 0 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

für die Veranschaulichung von Standardinhalten des Stoffgebietes Analytische Geometrie in Kursen erleichtern, in denen diese Software ansonsten nur sporadisch genutzt und häufig mit dem Lehrbuch gearbeitet wird. Der Nachteil dieser Variante besteht darin, dass die in den Makros definierten Objekte nicht mit den in POV-Ray enthaltenen Objekten kombiniert werden können, da deren Anordnung anhand des linkshändigen Koordinatensystems erfolgt. Im Folgenden sind die Makros der Datei `anageoR.inc` angegeben, diejenigen in `anageoL.inc` sind bis auf die dort nicht vorgenommenen Koordinatentransformationen identisch aufgebaut.

```
// -----
// anageoR.inc: Makros für die Darstellung von Vektoren, Geraden, Ebenen, ...
// (mit Transformation in rechtshändiges Koordinatensystem)
// -----
// Darstellung von Punkten (als kleine Kugeln)
// -----
#macro punkt (Q, textur)
    #declare QTRANS = <Q.y,Q.z,-Q.x>;
    sphere{<QTRANS.x,QTRANS.y,QTRANS.z>, intervall/40 texture{textur}}
#end
```

```
// Makro für die Darstellung von Verbindungsvektoren zwischen zwei Punkten
// -----
#macro verbindungsvektor (R, S , textur)
  #declare K=<R.y,R.z,-R.x>;
  #declare L=<S.y,S.z,-S.x>;
  #declare V=L-K;
  #declare vektorbetrag=sqrt(V.x*V.x+V.y*V.y+V.z*V.z);
  #if ( V.x*V.x+V.y*V.y = 0 )
    #declare PFEIL =
      union {cylinder{<0,0,0> <0,0,vektorbetrag-0.15*intervall> intervall/60}
        cone{<0,0,vektorbetrag-0.15*intervall> intervall/30
          <0,0,vektorbetrag> 0 }
          texture {textur} no_shadow
        };
    object { PFEIL translate <K.x,K.y,K.z> }
  #else
    #declare PFEIL =
      union {cylinder{<0,0,0> <vektorbetrag-0.15*intervall,0,0> intervall/60}
        cone{ <vektorbetrag-0.15*intervall,0,0> intervall/30
          <vektorbetrag,0,0> 0 }
          texture {textur} no_shadow
        };
    #declare NX=vnormalize(V);
    #declare NY=vnormalize(vcross(NX,z));
    #declare NZ=vnormalize(vcross(NX,NY));
    object { PFEIL matrix < NX.x , NX.y , NX.z ,
                          NY.x , NY.y , NY.z ,
                          NZ.x , NZ.y , NZ.z ,
                          K.x , K.y , K.z > }

  #end
#end
// -----
// Makro für die Darstellung von Ortsvektoren
// -----
#macro ortsvektor (O , textur)
  verbindungsvektor (<0,0,0>, O , textur)
#end
// -----
// Makro für die Darstellung von Pfeilen mit beliebigen Anfangspunkten
// -----
#macro vektoranpunkt (Punkt, Vektor , textur)
  verbindungsvektor (Punkt, Punkt + Vektor , textur)
#end
// -----
// Makro für die Darstellung von Strecken (als Zylinder mit geringem Radius)
// -----
#macro strecke (I, J , textur)
```

```

#declare RTRANS=<I.y,I.z,-I.x>;
#declare STRANS=<J.y,J.z,-J.x>;
#declare V=STRANS-RTRANS;
#if ( V.x*V.x+V.y*V.y+V.z*V.z = 0 )
#else cylinder { <I.y,I.z,-I.x> <J.y,J.z,-J.x> intervall/115
                texture {textur} no_shadow
            }
#end
#end
// -----
// Makro für die Darstellung von Geraden (als sehr lange Zylinder)
// -----
#macro gerade (R, S , textur)
#declare PP=<R.y,R.z,-R.x>;
#declare QQ=<S.y,S.z,-S.x>;
#declare V=QQ-PP;
#declare vektorbetrag=sqrt(V.x*V.x+V.y*V.y+V.z*V.z);
#if ( V.x*V.x+V.y*V.y = 0 )
#declare Streckenzylinder =
    cylinder { <0,0,-100*intervall> <0,0,100*intervall> intervall/150
              texture {textur} no_shadow
    };
    object { Streckenzylinder translate <PP.x,PP.y,PP.z> }
#else
#declare Streckenzylinder =
    cylinder { <-100*intervall,0,0> <100*intervall,0,0> intervall/150
              texture {textur} no_shadow
    };
#declare NX=vnormalize(V) ;
#declare NY=vnormalize(vcross(NX,z)) ;
#declare NZ=vnormalize(vcross(NX,NY)) ;
    object { Streckenzylinder matrix < NX.x , NX.y , NX.z ,
                                         NY.x , NY.y , NY.z ,
                                         NZ.x , NZ.y , NZ.z ,
                                         PP.x , PP.y , PP.z > }

#end
#end
// -----
// "pluspunkt": Punkt mit Loten auf die x-y-Ebene und die Achsen
// -----
# declare hellgrau = {texture { pigment { rgbf<0.6,0.6,0.6,0.9> }
                        finish { ambient 0.7 diffuse 0.5 roughness 0.1 }}};
#macro pluspunkt (PK, textur)
    sphere{<PK.y,PK.z,-PK.x>, intervall/40 texture{textur}}
    strecke( <PK.x,PK.y,PK.z>, <PK.x,PK.y,0>, texture{hellgrau} )
    strecke( <0,PK.y,0>, <PK.x,PK.y,0>, texture{hellgrau} )
    strecke( <PK.x,PK.y,0>, <PK.x,0,0>, texture{hellgrau} )
#end

```

```
// Makro für die Erzeugung von "Ebenen" als dünne Quader aus Gleichungen
// in drei Variablen (Ax+By+Cz+D=0)
// -----
#macro ebene (A,B,C,D,textur)
  #if ( B = 0 )
    #if ( C = 0 )
      box {<-intervall/1000, -intervall, -intervall>
        <intervall/1000, intervall, intervall>
        translate <-D/A,0,0>   matrix < 0,0,-1,  1,0,0,  0,1,0,  0,0,0 >
        texture {textur}   no_shadow
      }
    #else
      box {<-intervall, -intervall, -intervall/1000>
        <intervall, intervall, intervall/1000>
        rotate <0, degrees(atan2(A,C)) ,0>
        translate <0,0,-D/C>   matrix < 0,0,-1,  1,0,0,  0,1,0,  0,0,0 >
        texture {textur}   no_shadow
      }
    #end
  #else
    #if ( A*A + C*C = 0 )
      box {<-intervall, - intervall/1000, -intervall>
        <intervall, intervall/1000, intervall>
        translate <0,-D/B,0>   matrix < 0,0,-1,  1,0,0,  0,1,0,  0,0,0 >
        texture {textur}   no_shadow
      }
    #else
      #declare VX1 = <0,1,0>;
      #declare VX2=vnormalize(<A,B,C>);
      #declare VY=vnormalize(vcross(VX2,VX1));
      #declare VZ1=vcross(VY,VX1);
      #declare VZ2=vcross(VY,VX2);
      box {<-intervall, -intervall/1000, -intervall>
        <intervall, intervall/1000, intervall>
        matrix < VX1.x, VY.x, VZ1.x,
                  VX1.y, VY.y, VZ1.y,
                  VX1.z, VY.z, VZ1.z,
                  0      0      0 >
        matrix < VX2.x, VX2.y, VX2.z,
                  VY.x,  VY.y,  VY.z,
                  VZ2.x, VZ2.y, VZ2.z,
                  0,      0,      0 >
        translate <0,-D/B,0>   matrix < 0,0,-1,  1,0,0,  0,1,0,  0,0,0 >
        texture {textur}   no_shadow
      }
    #end
  #end
#end
#end
#end
```



```
// Makro für die Erzeugung von "Ebenen" aus Parameterdarstellungen
```

```
// -----
```

```
#macro ebenepar (M0,m,n,textur)
  #declare P= <M0.y,M0.z,-M0.x>;
  #declare a= <m.y,m.z,-m.x>;
  #declare b= <n.y,n.z,-n.x>;
  #declare normvekt = vcross(a,b);
  #declare A = normvekt.x;
  #declare B = normvekt.y;
  #declare C = normvekt.z;
  #if ( B = 0 )
    #if ( C = 0 )
      box {<-intervall/1000, -intervall, -intervall>
        <intervall/1000, intervall, intervall>
        translate P texture {textur} no_shadow }
    #else
      box {<-intervall, -intervall, -intervall/1000>
        <intervall, intervall, intervall/1000>
        rotate <0, degrees(atan2(A,C)), 0> translate P
        texture {textur} no_shadow}
    #end
  #else
    #if ( A*A + C*C = 0 )
      box {<-intervall, -intervall/1000, -intervall>
        <intervall, intervall/1000, intervall>
        translate P texture {textur} no_shadow }
    #else
      #declare VX1 = <0,1,0>;
      #declare VX2=vnormalize(<A,B,C>);
      #declare VY=vnormalize(vcross(VX2,VX1));
      #declare VZ1=vcross(VY,VX1);
      #declare VZ2=vcross(VY,VX2);
      box {<-intervall, -intervall/1000, -intervall>
        <intervall, intervall/1000, intervall>
        matrix < VX1.x, VY.x, VZ1.x,
                  VX1.y, VY.y, VZ1.y,
                  VX1.z, VY.z, VZ1.z,
                  0 0 0 >
        matrix < VX2.x, VX2.y, VX2.z,
                  VY.x, VY.y, VY.z,
                  VZ2.x, VZ2.y, VZ2.z,
                  0, 0, 0 >
        translate P texture {textur} no_shadow }
    #end
  #end
#end
```



# Anhang B

## Farbige Abbildungen

Dieser Anhang enthält farbige Versionen einiger Abbildungen der Arbeit, bei denen durch die Umwandlung in Graustufenbilder wichtige Informationen bzw. Eindrücke verloren gegangen sind, sowie Arbeiten von Schülern.

### Abbildungen zu Kapitel 2

#### Abschnitt 2.1

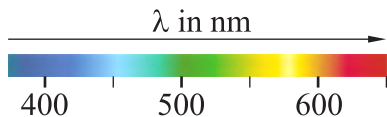


Abbildung B.1: Spektrum des sichtbaren Lichts

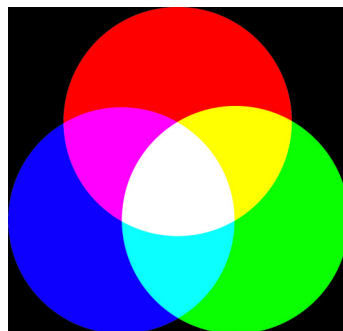


Abbildung B.2: Additive Farbmischung

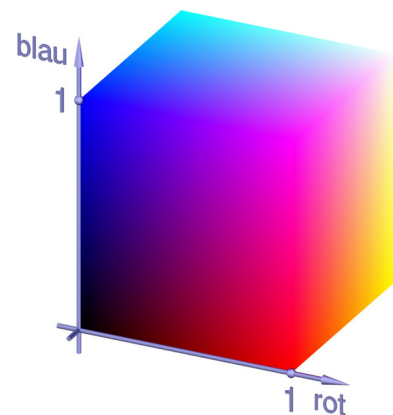


Abbildung B.3: RGB-Würfel

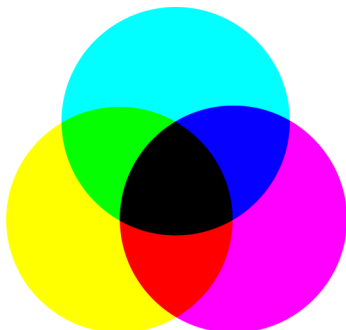


Abbildung B.4: Subtraktive Farbmischung

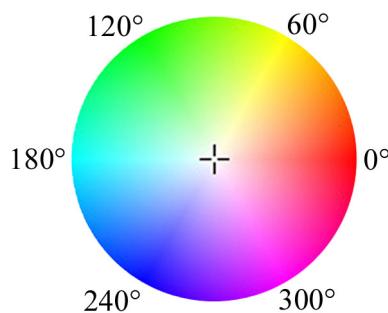


Abbildung B.5: Farbkreis

### Abbildung zu Abschnitt 2.2

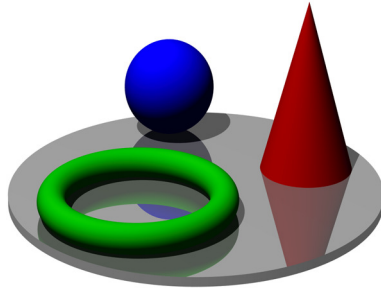


Abbildung B.6: Beispiel einer dreidimensionalen Computergrafik

### Abbildungen zu Abschnitt 2.3

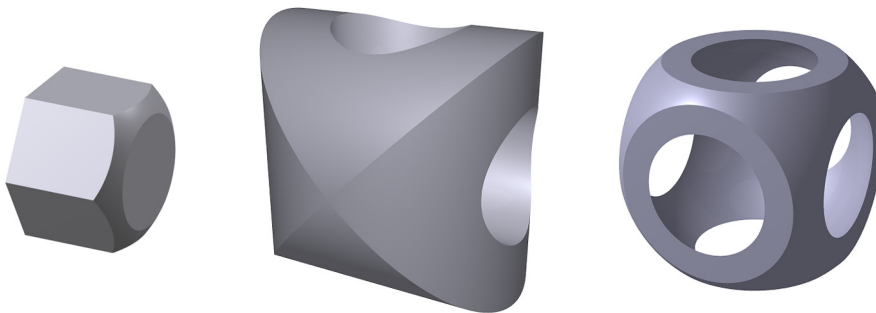


Abbildung B.7:  
Boolesche  
Operationen

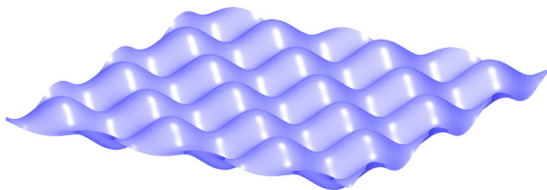


Abbildung B.8: Graph einer Funktion zweier Variablen

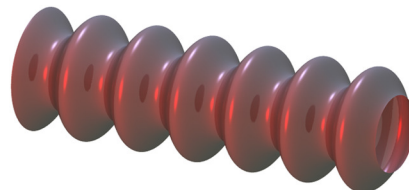


Abbildung B.9: Darstellung einer Fläche durch eine implizite Gleichung

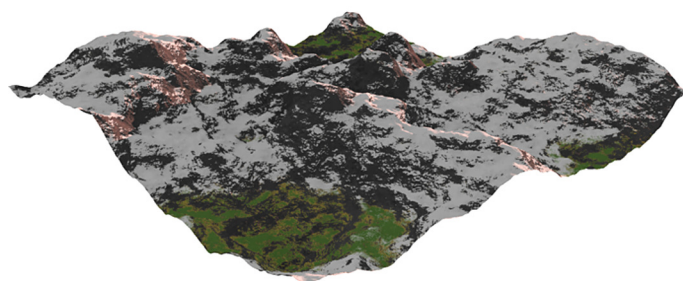
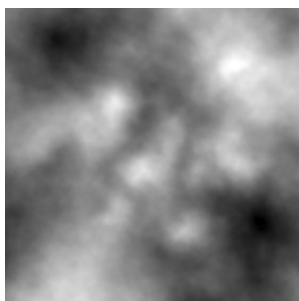


Abbildung B.10: Durch Höhenfunktion aus einem Graustufenbild erzeugtes Terrain

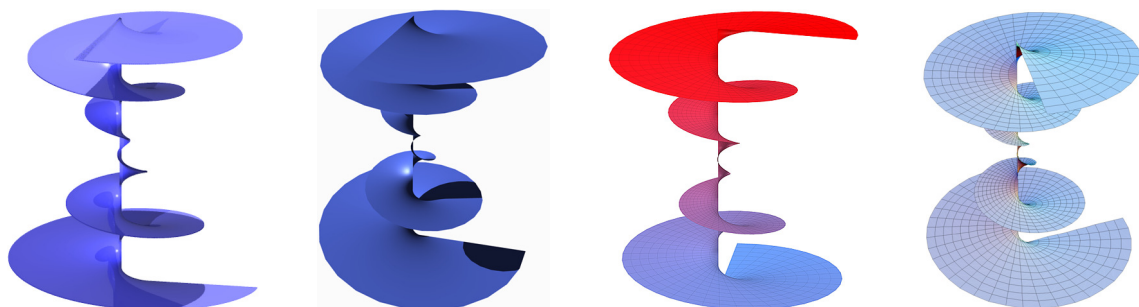


Abbildung B.11: Darstellung einer durch eine Parameterdarstellung gegebenen konischen Spiralfäche in POV-Ray, Carrara, MuPAD und Mathematica (v. l. n. r.)

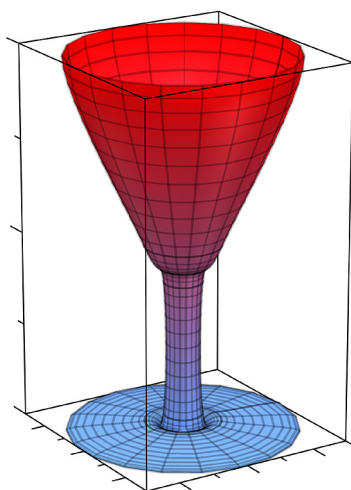


Abbildung B.12: Darstellung einer Rotationsfläche in MuPAD und POV-Ray

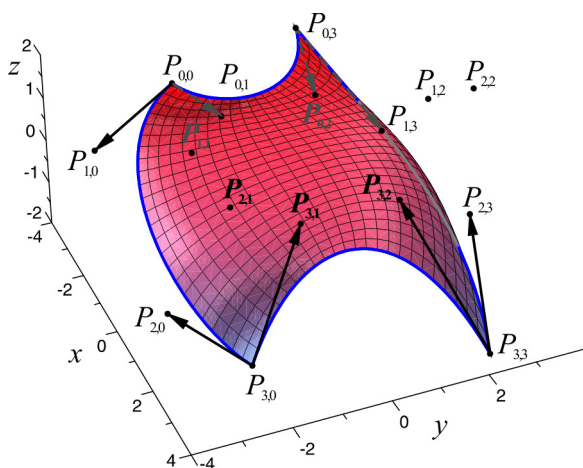


Abbildung B.13: Bézierflächenstück

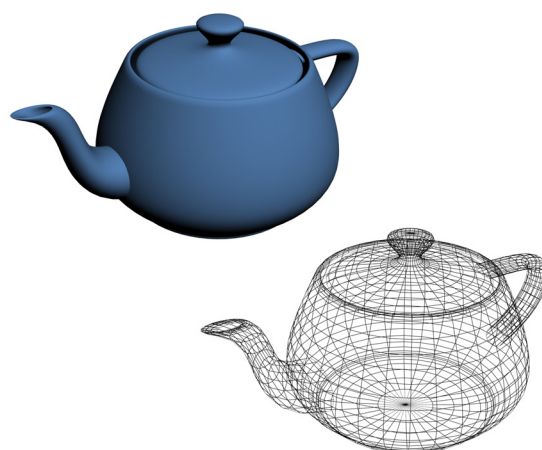


Abbildung B.14: Utah Teapot

## Abbildungen zu Abschnitt 2.4

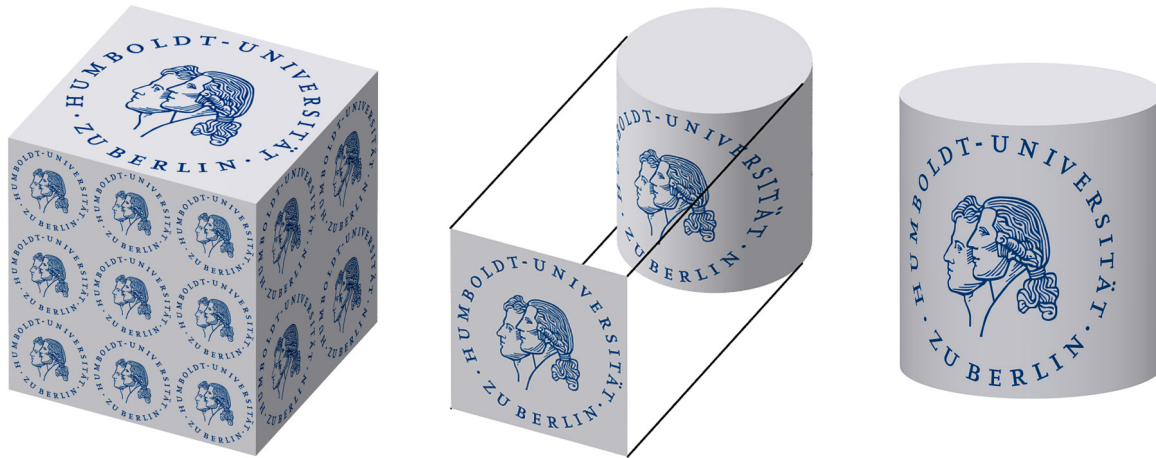


Abbildung B.15: Ebene (links und Mitte) sowie zylindrische Texturprojektion (rechts)



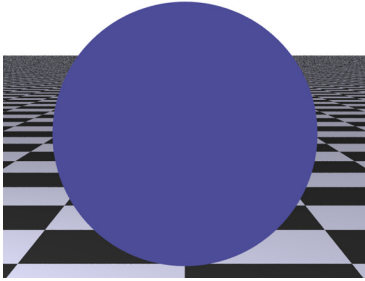
Abbildung B.16: Sphärische Texturprojektion



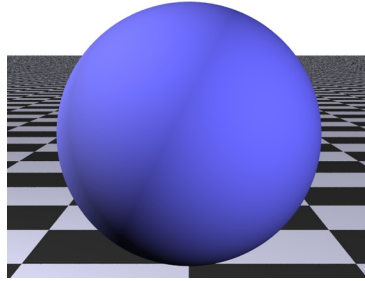
Abbildung B.17: UV-Mapping



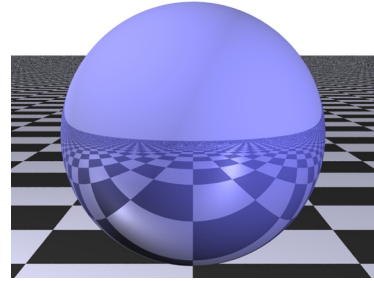
## Abbildungen zu Abschnitt 2.5



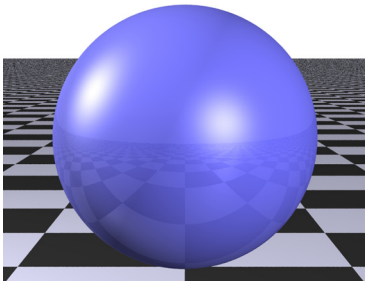
a)  $k_a = 0,9$ ;  $k_d = 0$ ;  
 $k_s = 0$ ;  $k_t = 0$ ;  
 $k_p = 0$



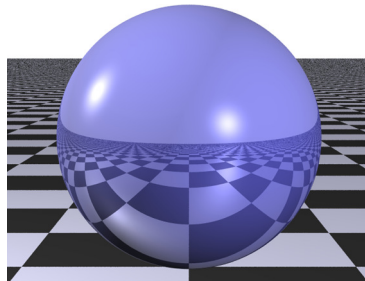
b)  $k_a = 0,1$ ;  $k_d = 1$ ;  
 $k_s = 0$ ;  $k_t = 0$ ;  
 $k_p = 0$



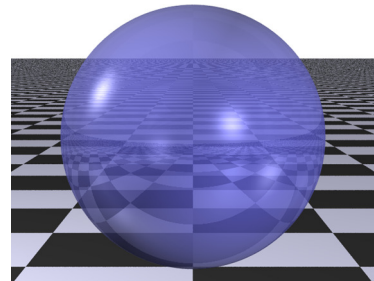
c)  $k_a = 0,1$ ;  $k_d = 0,7$ ;  
 $k_s = 0,3$ ;  $k_t = 0$ ;  
 $k_p = 0$



d)  $k_a = 0,1$ ;  $k_d = 0,7$ ;  
 $k_s = 0,1$ ;  $k_t = 0$ ;  
 $k_p = 0,5$ ;  $m = 12$



e)  $k_a = 0,1$ ;  $k_d = 0,7$ ;  
 $k_s = 0,3$ ;  $k_t = 0$ ;  
 $k_p = 0,5$ ;  $m = 50$



f)  $k_a = 0,1$ ;  $k_d = 0,7$ ;  
 $k_s = 0,3$ ;  $k_t = 0,4$ ;  
 $k_p = 0,5$ ;  $m = 50$

Abbildung B.18: Wirkung verschiedener Kombinationen der Parameter **ambient**  $k_a$ , **diffuse**  $k_d$ , **reflection**  $k_s$ , **phong**  $k_p$ , **phong\_size**  $m$  sowie der Transparenz  $k_t$  auf das Erscheinungsbild einer Kugel.

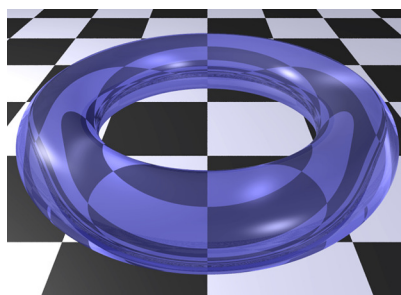
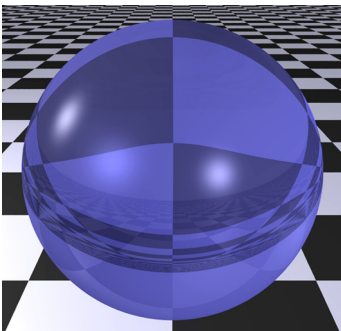


Abbildung B.19: Gefilterte Transparenz und Brechung (Brechungsindex **ior** = 1,5)

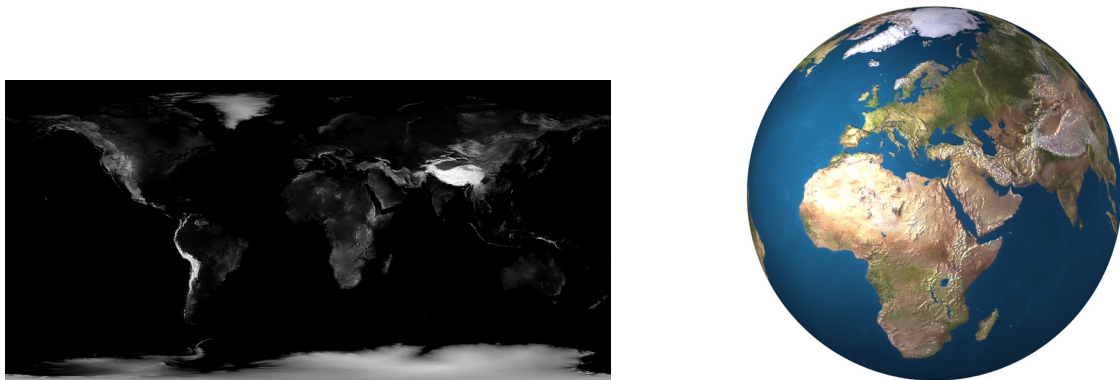


Abbildung B.20: Kugel mit Bump Map (Höhenprofil) der Erde

Die rechte Abb. entspricht – bis auf die zusätzlich verwendete (links abgebildete) Bump Map – der Abb. der Erde auf S. 352.

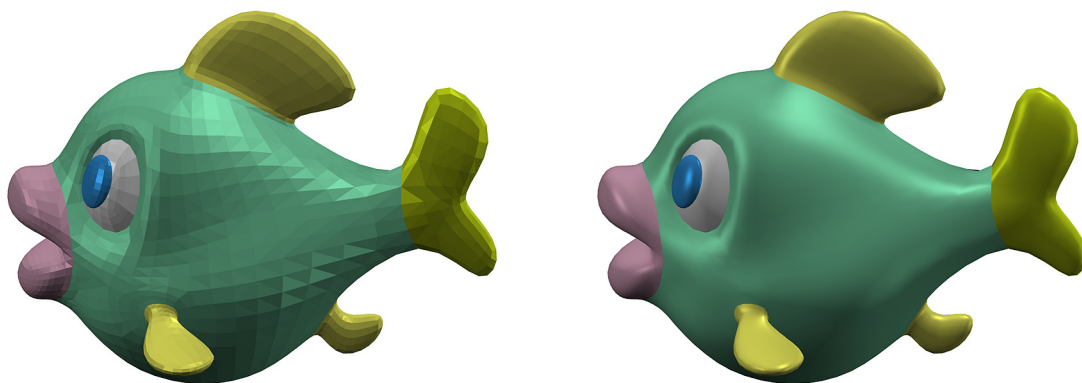


Abbildung B.21: Flat Shading (links), Phong Shading (rechts)

Beide Abbildungen wurden unter Verwendung desselben Dreiecksnetzes erzeugt. Der rechts abgebildete Fisch besteht somit aus genau so vielen Eckpunkten und Dreiecken wie der links dargestellte Fisch. Der Unterschied kommt allein durch die Normaleninterpolation zustande.

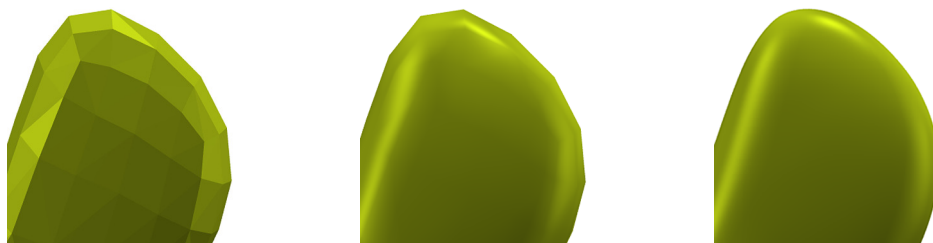


Abbildung B.22: Vergleich: Flat Shading, Phong Shading, Subdivision Surface  
(Ausschnitt der Schwanzflosse des oben abgebildeten Fisches)



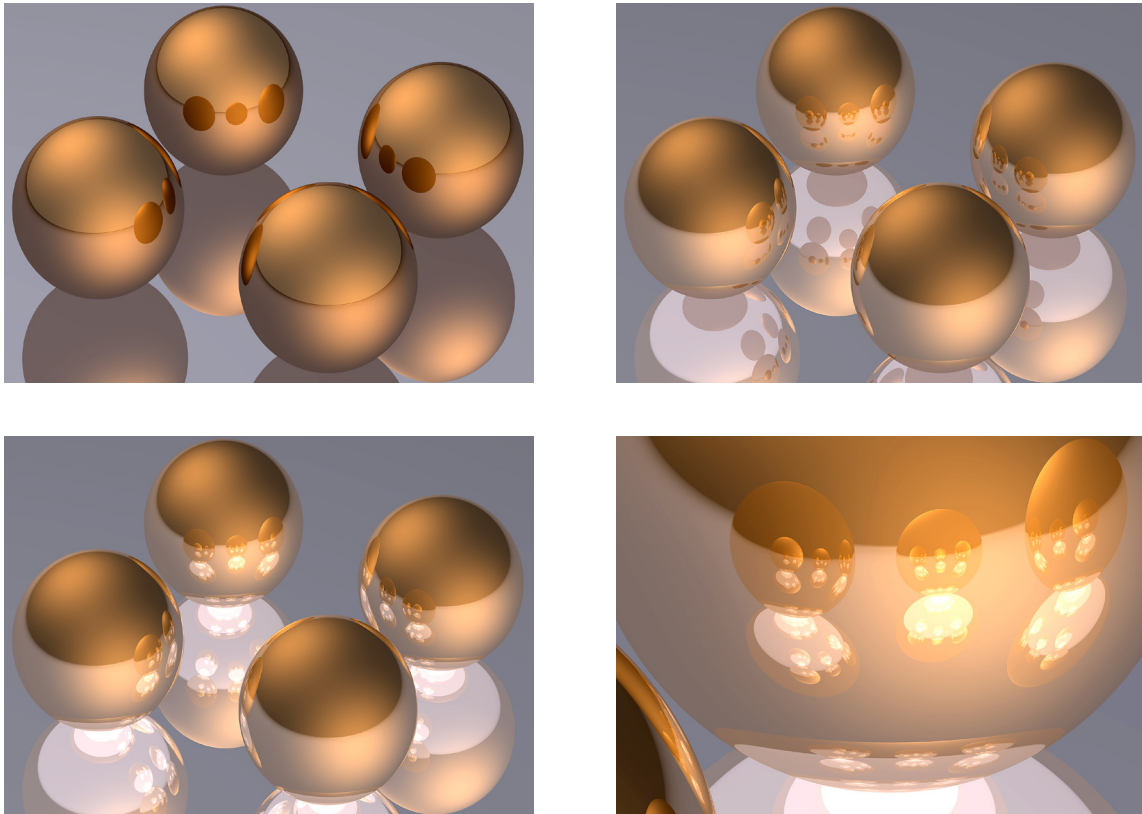


Abbildung B.23: Mehrfachreflexionen – Einfluss der Strahlverfolgungstiefe

Alle vier Abbildungen zeigen dieselbe Szene, die aus vier Kugeln und einer Ebene besteht, auf der die Kugeln liegen; sie unterscheiden sich nur in der Strahlverfolgungstiefe (links oben: 2, rechts oben: 4, unten: 20). Unten rechts ist ein Ausschnitt des Bildes unten links abgebildet.

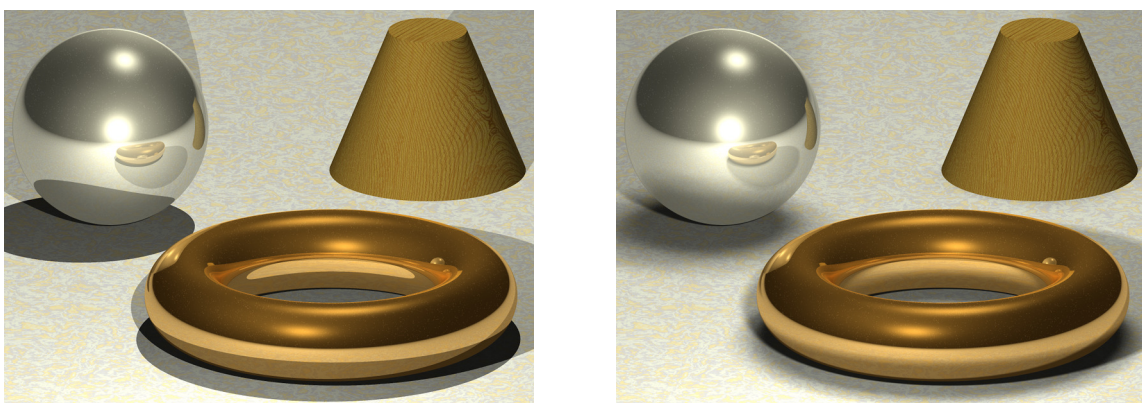


Abbildung B.24: Szene mit punktförmigen (li.) und ausgedehnten Lichtquellen (re.)

Die beiden Lichtquellen in der rechten Abb. bestehen jeweils aus  $6 \cdot 6$  Punktlichtern. Die Renderzeit der linken Abbildung betrug 9 Sekunden, die der rechten Abbildung 1 Minute und 22 Sekunden (Pentium IV, 3,2 GHz, Auflösung jeweils  $800 \times 600$  Pixel mit Anti-Aliasing).

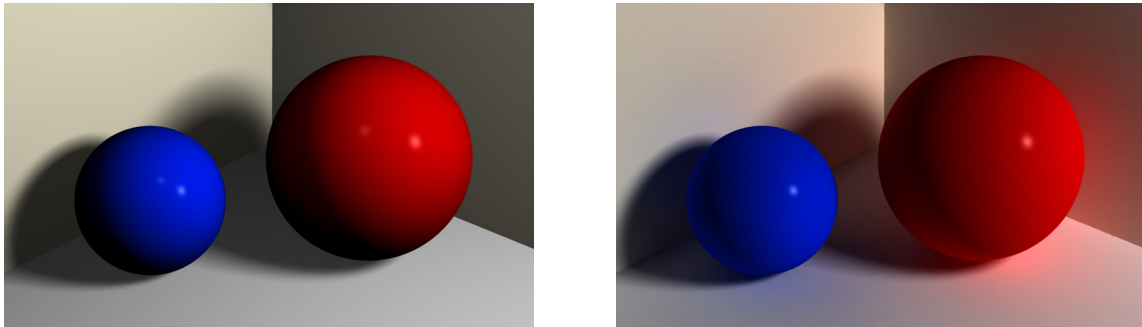


Abbildung B.25: Szene ohne (links) und mit Radiosity (rechts)

Um den Unterschied zwischen der Szene mit und ohne Radiosity trotz drucktechnischer Unzulänglichkeiten hinreichend deutlich werden zu lassen, wurden recht hohe Werte für den Radiosity-Anteil eingestellt. Für einen realistischeren Eindruck bei der Darstellung auf Bildschirmen sollten geringere Werte gewählt werden (siehe die Abb. **Szene-mit-Radiosity.jpg** in dem Ordner 2-5).

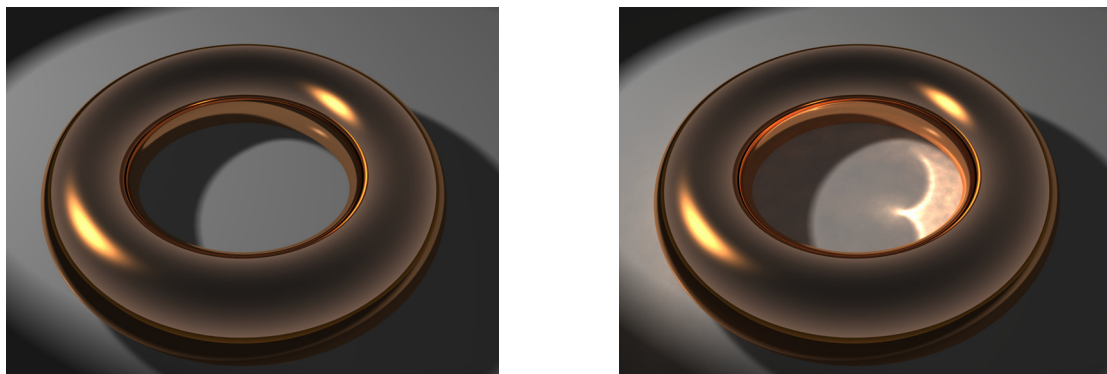


Abbildung B.26: Szene ohne (links) und mit Caustics (rechts)

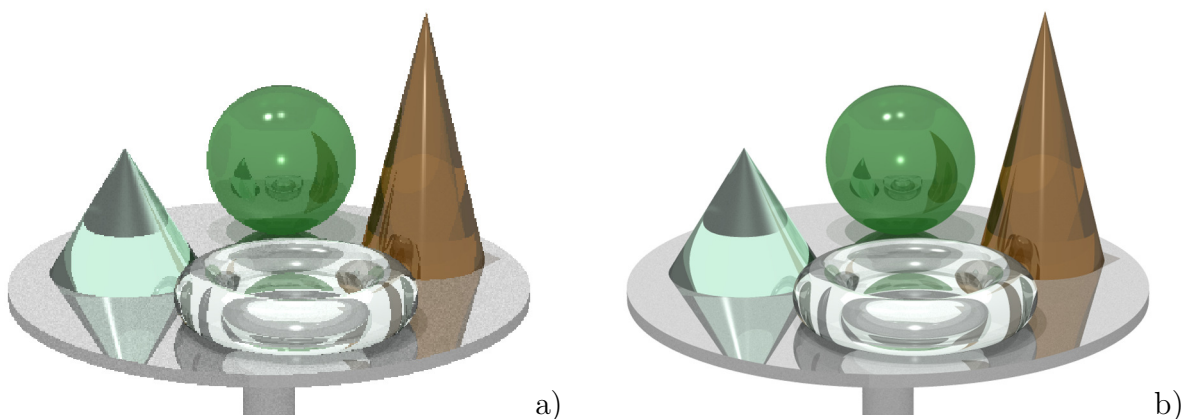


Abbildung B.27: Einfluss von Anti-Aliasing auf die Bildqualität

Die Bildberechnung der Szene in POV-Ray benötigte mit einer Auflösung von jeweils  $800 \times 600$  Pixeln 9 Sekunden ohne und 23 Sekunden mit Anti-Aliasing.

## Von Schülern angefertigte Abbildungen

### Schneemänner

Die Abbildungen auf dieser und der folgenden Seite wurden von Schülern eines Mathematik-Grundkurses des Andreas-Gymnasiums in Berlin-Friedrichshain im Herbst 2003 angefertigt (vgl. die Abschnitte 4.1 und 5.1). Die von den Schülern erstellten POV-Ray-Dateien mit den Beschreibungen der Schneemänner befinden sich in dem Ordner `Schuelerarbeiten\Schneemaenner-AGym-Berlin` (siehe Anhang D, S. 365).

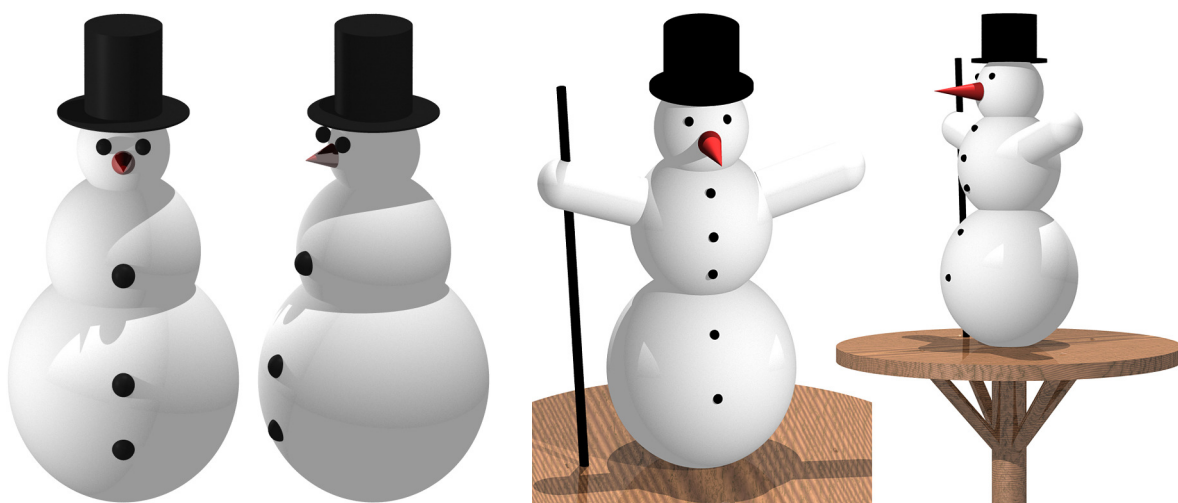


Abbildung B.28: Von Schülern eines Grundkurses mit POV-Ray angefertigte Szenen

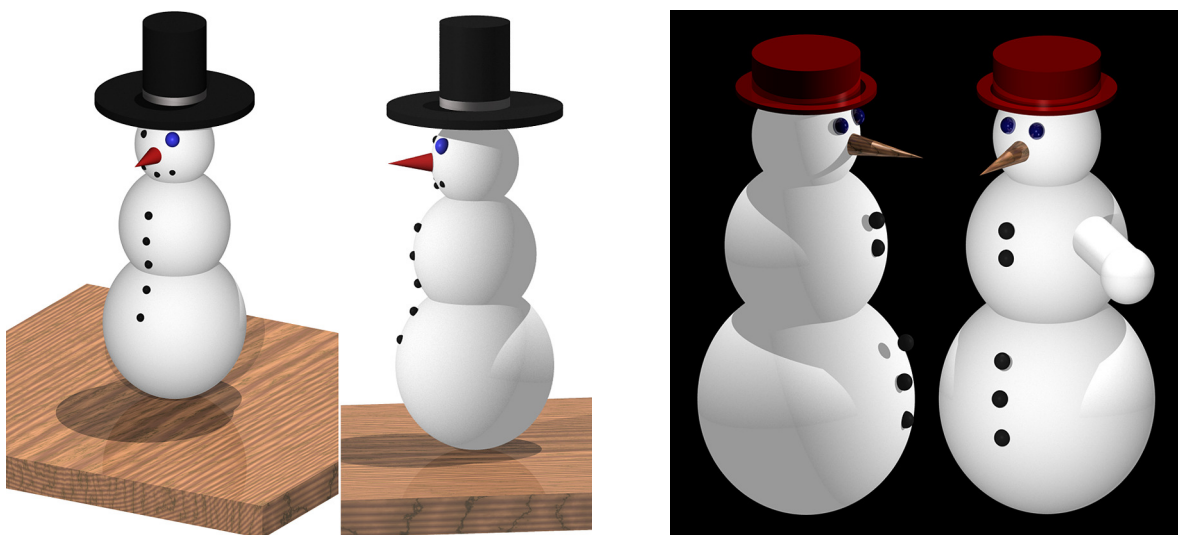


Abbildung B.29: Von Schülern eines Grundkurses mit POV-Ray angefertigte Szenen



Abbildung B.30: Von Schülern eines Grundkurses mit POV-Ray angefertigte Szenen



Abbildung B.31: Von Schülern eines Grundkurses mit POV-Ray angefertigte Szenen



Die folgenden Abbildungen wurden von Schülern einer 12. Klasse der Internationalen Gesamtschule Heidelberg (IGH) im Frühsommer 2005 innerhalb einer Unterrichtsreihe angefertigt, die von J. KÖBERLE durchgeführt wurde (siehe die Abschnitte 4.1 und 5.2). Die von den Schülern dazu erstellten POV-Ray-Dateien befinden sich in dem Ordner `Schuelerarbeiten\Schneemaenner-IGH-Heidelberg` (siehe S. 365). Die Prüfungsarbeit von Herrn KÖBERLE ist in dem Ordner `Pruefungsarbeit-J-Koeberle` enthalten. Herr KÖBERLE hat ausdrücklich der Veröffentlichung seiner Arbeit und der Schülerdateien innerhalb der vorliegenden Arbeit zugestimmt.

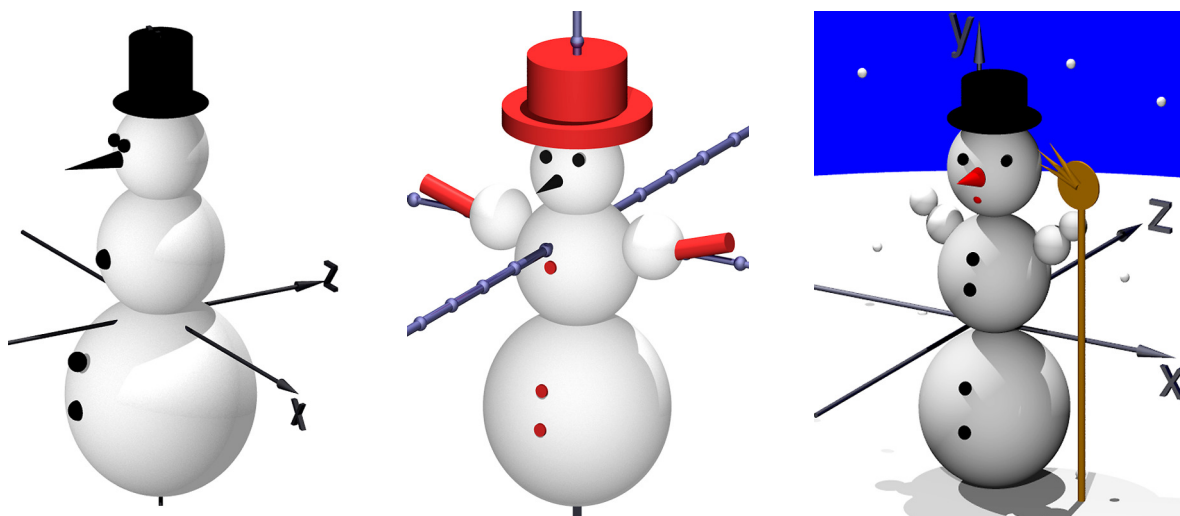


Abbildung B.32: Von Schülern einer 12. Klasse mit POV-Ray angefertigte Szenen

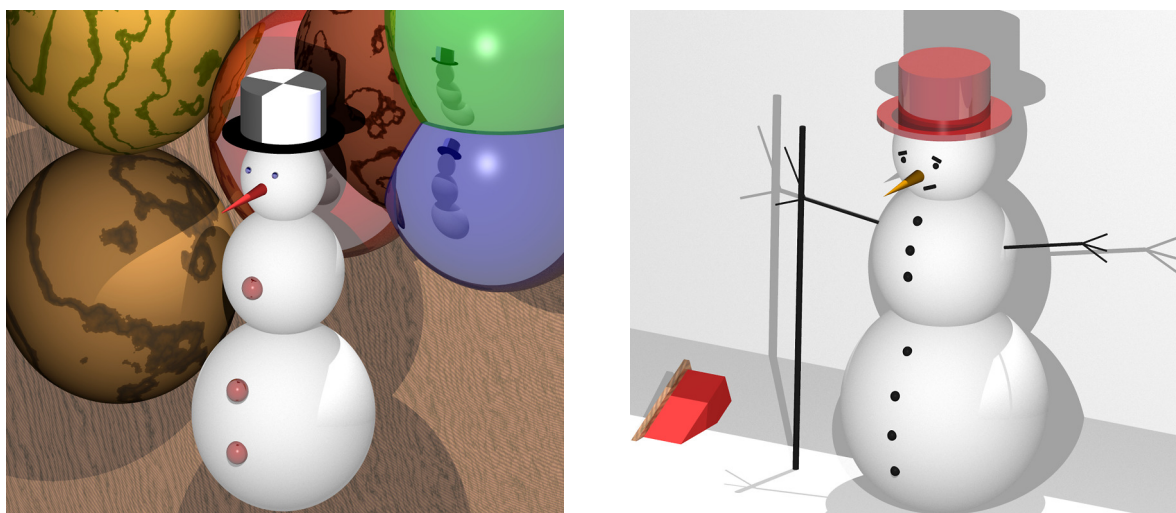


Abbildung B.33: Von Schülern einer 12. Klasse mit POV-Ray angefertigte Szenen

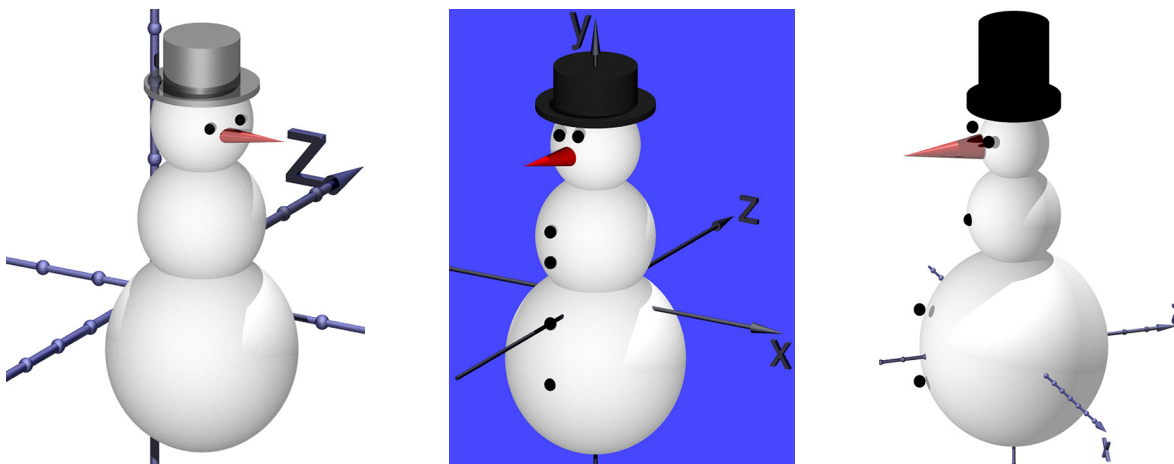


Abbildung B.34: Von Schülern einer 12. Klasse mit POV-Ray angefertigte Szenen

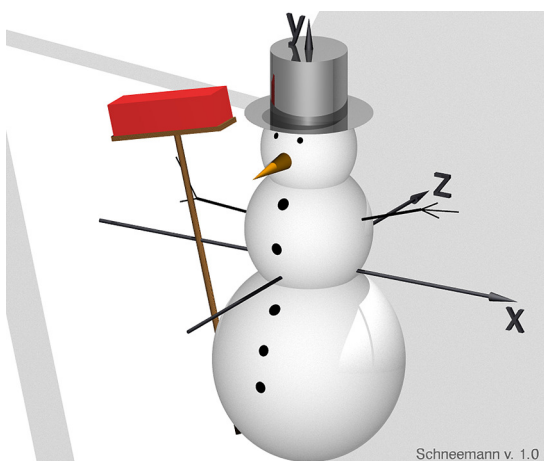


Abbildung B.35: Von einem Schüler einer 12. Klasse mit POV-Ray angefertigte Szene in drei Ansichten

## Ergebnisse aus einem Unterrichtsprojekt in einem Grundkurs

Die folgenden Abbildungen wurden von Schülern eines Grundkurses ma-13 des Fürst-Pückler-Gymnasiums in Cottbus im November 2002 innerhalb eines dreiwöchigen Unterrichtsprojektes angefertigt, das von F. RIEPER durchgeführt wurde (siehe dazu Abschnitt 5.3). Da viele Schüler in diesem Projekt Videos erstellten, sind hier Einzelbilder aus einigen dieser Videos abgedruckt. Die Videos und die zugehörigen POV-Ray-Dateien der Schüler mit den Szenenbeschreibungen befinden sich in dem Ordner `Schuelerarbeiten\Projekt-FPG-Cottbus-ma13-F-Rieper` (siehe S. 365).

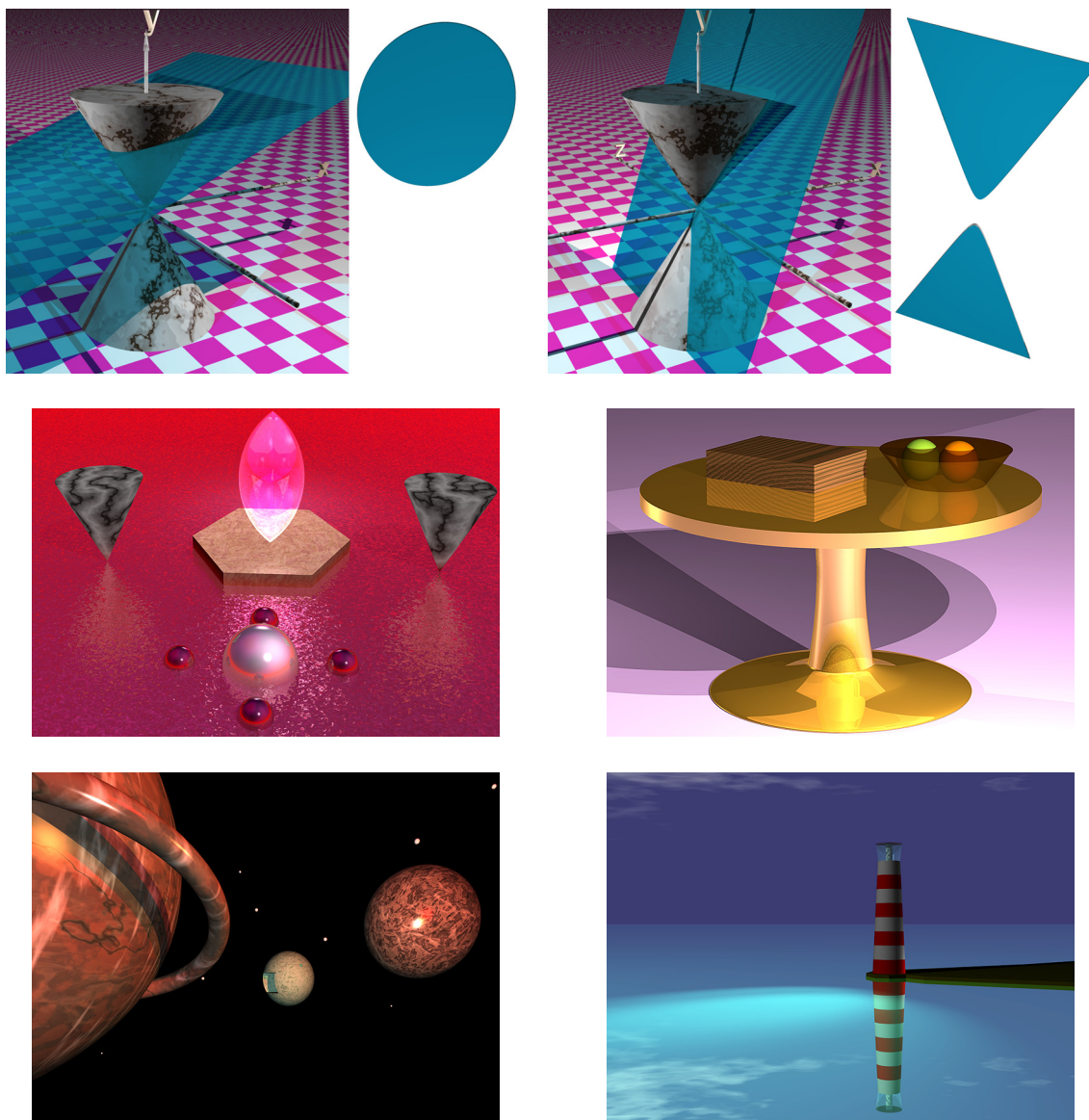


Abbildung B.36: Von Schülern eines Grundkurses mit POV-Ray angefertigte Szenen



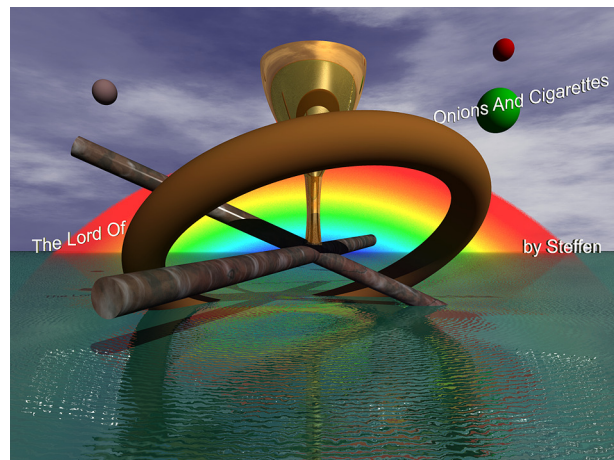
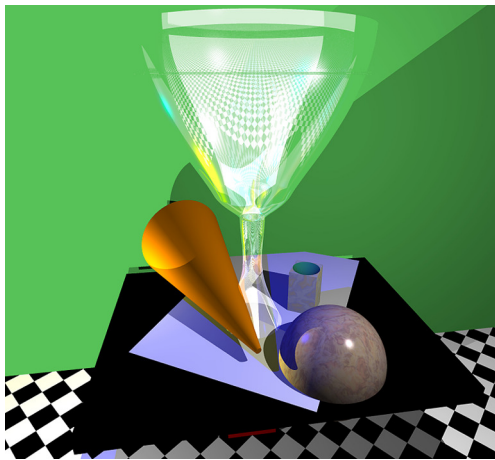


Abbildung B.37: Von Schülern eines Grundkurses mit POV-Ray angefertigte Szenen



Abbildung B.38: Von einer Schülerin eines Grundkurses mit POV-Ray angefertigte Szene "Mittelalterliche Tafel" in drei Ansichten. Die Schülerin erstellte dazu eine besonders aufwändige Animation (Ordner Schuelerarbeiten\Projekt-FPG-Cottbus-ma13-F-Rieper\Juliane-Mittelalter).



## Nachweis der Bildquellen

Alle Abbildungen, die auf den Seiten 349–356 dieses Anhangs abgedruckt sind, wurden von dem Autor der vorliegenden Arbeit angefertigt, wobei folgende (lizenzfrei nutzbare) Ressourcen Verwendung fanden:

- Beschreibung der Utah-Teekanne als Bézierpatch für Abb. B.14 auf S. 351 und Abb. B.17 auf S. 352 (verfügbar unter [www.povray.org](http://www.povray.org), auf diversen anderen Internetseiten und auf Begleit-CDs zu Fachbüchern);
- Logo der Humboldt-Universität zu Berlin für Abb. B.15 und B.17 auf S. 352: <http://zope.hu-berlin.de/hu-intern/design/cd/logo/>;
- Karte (quadratische Plattkarte) der Erde für Abb. B.16 auf S. 352 sowie zusätzlich eine Bump Map der Erde für Abb. B.20 auf S. 354 von XGlobe & XPlanet Maps: <http://www.radcyberzine.com/xglobe>;
- Dreiecksnetz eines Fisches (als .3ds-Datei) für Abb. B.21 und B.22 auf S. 354 von <http://www.3dcafe.com>.

Zusätzlich wurden für die Graustufenabbildungen in Kapitel 2 der vorliegenden Arbeit die folgenden, ebenfalls lizenzfrei nutzbaren, Dateien verwendet:

- ein Bild eines Fisches von der mit der Software Corel Draw 12 gelieferten Content-CD für Abb. 2.2 auf S. 68,
- ein hoch aufgelöstes Dreiecksnetz eines Frosches für Abb. 2.43 auf S. 110 von der CD „500 3D Objects“ (Taschen Verlag, 2002); die Nutzung und Bearbeitung der Objekte auf dieser CD ist nach den Lizenzbestimmungen des Verlages erlaubt.

Die Abbildungen ab S. 357 dieses Anhangs haben Schülerinnen und Schülern in Kursen erstellt, die vom Autor selbst sowie von J. KÖBERLE und F. RIEPER geleitet wurden. Diese beiden Kollegen haben die Veröffentlichung der Abbildungen und der zugehörigen Dateien im Rahmen der vorliegenden Arbeit ausdrücklich gestattet. Auch die Schüler, von denen die Abbildungen angefertigt wurden, haben sich mit einer Veröffentlichung einverstanden erklärt.

Alle anderen in dieser Arbeit enthaltenen Abbildungen – mit wenigen Ausnahmen, die jeweils explizit gekennzeichnet sind – hat der Autor angefertigt. Es werden somit keine Rechte Dritter verletzt.

# Anhang C

## Notation

$n, m, i, j$	–	ganzzahlige Variable, Indizes
$a, b, x, y$	–	reellwertige Variable
$A, B, P, Q$	–	Punkte
$P(x; y; z)$	–	Punkt mit Koordinatenangaben
$(x; y; z; w)$	–	homogene Koordinaten
$\overline{AB}, \overline{PQ}$	–	Verbindungsstrecken von Punkten
$g, AB$	–	Gerade, Gerade durch zwei Punkte
$\vec{n}, \vec{p}, \vec{x}$	–	Vektoren
$\overrightarrow{AB}, \overrightarrow{PQ}$	–	Verbindungsvektoren von Punkten
$\begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}, (a_1, a_2, \dots, a_n)^T$	–	Vektoren in Komponentenschreibweise
$\langle \vec{p}, \vec{q} \rangle$	–	Skalarprodukt zweier Vektoren
$\vec{p} \times \vec{q}$	–	Vektorprodukt zweier Vektoren
$A, (a_{ij})$	–	Matrizen
$A \cdot B$	–	Produkt zweier Matrizen
$(x_1; x_2)$	–	offenes Intervall
$[x_1; x_2]$	–	abgeschlossenes Intervall
$f(x), g(x)$	–	Funktionsterme reellwertiger Funktionen
$f(x, y), f(x, y, z)$	–	Funktionsterme von Funktionen mehrerer Variablen
$f'(x), f''(x)$	–	1. und 2. Ableitung einer Funktion
$\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$	–	partielle Ableitungen einer Funktion $f(x, y, z)$
$\vec{x}(t), \vec{x}(u, v)$	–	Vektorfunktionen (Parameterdarstellungen)
$\frac{dx}{dt}, \frac{dy}{dt}, \frac{dz}{dt}$	–	Ableitungen der Komponenten einer Vektorfunktion
$\vec{x}(t) = \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix}$	–	nach $t$

# Anhang D

## Hinweise zu den Dateien auf der Internetseite zu dieser Arbeit

Auf der Internetseite <http://www.afiller.de/habil> stehen Beispieldateien zu Abschnitten der Arbeit, ergänzende Materialien sowie von Schülern in Unterrichtsversuchen und von Studierenden angefertigte Dateien zur Verfügung. Sie sind in einem Archiv (.zip-Datei: **Filler-Habil-Dateien.zip**) gepackt. Nach dem Herunterladen und Entpacken der .zip-Datei wird ein Ordner

### **Filler-Habil-Dateien**

mit folgenden Unterordnern angelegt:

#### **2-1 ... 4-6**

Diese Ordner enthalten Dateien, die sich unmittelbar auf Abschnitte der Arbeit beziehen (Ordner **n-m** enthält Dateien zu Abschnitt *n.m*).

#### **Pruefungsarbeit-J-Koeberle**

In diesem Ordner befindet sich die Prüfungsarbeit zum 2. Staatsexamen von J. KÖBERLE (siehe Abschnitt 5.2). Herr KÖBERLE stimmte der Veröffentlichung zu.

#### **Rahmenplaene**

Dieser Ordner enthält die im Zeitraum der Erstellung dieser Arbeit gültigen Rahmenpläne einer Reihe von Bundesländern für das Stoffgebiet Analytische Geometrie.

#### **Schuelerarbeiten**

Der Ordner enthält Dateien, die Schüler im Rahmen von Unterrichtsversuchen anfertigten (siehe Kapitel 5), sowie die auf S. 256 und S. 299 erwähnte Facharbeit eines Schülers.

#### **Schulversuch-Andreas-Gym**

Dieser Ordner enthält Materialien zu dem in Abschnitt 5.1 beschriebenen Schulversuch, Schülerdateien sowie Kopien der Schülerantworten bei einem Test und einer Umfrage.

#### **Studierenden-Arbeiten**

Der Ordner enthält Arbeiten, die Studierende anfertigten (siehe Abschnitt 5.4).

#### **Vorlagen+Anleitungen**

In diesem Ordner befinden sich Kurzanleitungen und Vorlagen, die Schülern die Arbeit mit POV-Ray erleichtern sollen (siehe die Abschnitte 3.5 und 4.2 sowie Anhang A).

## Dateitypen

Die auf der vorherigen Seite kurz beschriebene Materialsammlung enthält mit verschiedenen Softwarepaketen erstellte Dateien. Im Folgenden werden die hauptsächlich vorhandenen Dateitypen anhand ihrer Extensionen kurz beschrieben.<sup>1</sup>

<b>.car</b>	Carrara-Szenendateien, die mit Carrara (ab Version 4) geöffnet werden können. Eine Testversion kann unter <a href="http://www.daz3d.com">http://www.daz3d.com</a> heruntergeladen werden.
<b>.inc</b> <b>.ini</b>	Include-Dateien (.inc), die von POV-Ray-Dateien aufgerufen werden (siehe die Hinweise unten), sowie Initialisierungsdateien (.ini) für die Erzeugung von Videos.
<b>.jpg</b> <b>.png</b> <b>.tif</b>	Bilddateien, die mit nahezu jeder Bildbetrachtungs- oder -bearbeitungssoftware geöffnet werden können.
<b>.mnb</b>	MuPAD-3-Notebooks, zum Öffnen ist MuPAD (Version 3.1 oder 3.2) erforderlich.
<b>.mn</b>	MuPAD-4-Notebooks, zum Öffnen ist MuPAD (ab Version 4) erforderlich. Eine 30 Tage lauffähige Testversion steht unter <a href="http://www.mupad.de">http://www.mupad.de</a> zur Verfügung.
<b>.mpg</b> <b>.avi</b>	Videodateien; zum Anschauen ist eine geeignete Abspielsoftware erforderlich (unter Windows z. B. der Windows Media Player; unter Mac OS z. B. Quicktime).
<b>.nb</b>	Mathematica-Notebooks, zum Öffnen ist Mathematica (ab Version 4) erforderlich.
<b>.pdf</b>	Dokumente und Beschreibungen im Portable Document Format; zum Lesen wird der Adobe Reader benötigt, der unter <a href="http://www.adobe.de">http://www.adobe.de</a> frei erhältlich ist.
<b>.pov</b>	POV-Ray-Dateien (vgl. die Hinweise unten). Mit den POV-Ray-Dateien befinden sich jeweils daraus berechnete Bilddateien (.jpg) in den entsprechenden Ordnern.
<b>.py</b>	Python-Skripte (siehe HINWEIS.txt in dem Ordner 2-6\Echtzeit-Animationen).
<b>.txt</b>	Textdateien (enthalten kurze Hinweise zu Inhalten entsprechender Ordner); können mit beliebigen Texteditoren geöffnet werden.

## Hinweise zur Verwendung von POV-Ray-Dateien

Viele der in der Materialsammlung (Filler-Habil-Dateien.zip) enthaltenen Dateien wurden mit der freien Raytracing-Software POV-Ray erstellt. Diese Dateien können mit POV-Ray (oder mit einem Texteditor) geöffnet werden.<sup>2</sup>

Für eine Reihe der bereitgestellten POV-Dateien werden Include-Dateien benötigt, die nach dem Entpacken der .zip-Datei in den entsprechenden Ordnern enthalten sind. Werden einzelne POV-Dateien ohne die benötigten Include-Dateien kopiert oder verschoben, so kann POV-Ray die Bildberechnung verweigern. Vor dem Verschieben oder Kopieren von POV-Dateien ist daher zunächst zu prüfen, welche Dateien diese mit **#include**-Befehlen aufrufen. *Am einfachsten ist es, die komplette Datei Filler-Habil-Dateien.zip zu entpacken und in den dabei entstandenen Ordnern zu arbeiten oder diese vollständig zu kopieren oder zu verschieben.*

Weitere Hinweise für die Verwendung von POV-Ray enthalten Anhang A und der Ordner Vorlagen+Anleitungen.

<sup>1</sup>In den Ordnern 2-1, 2-3 und 2-7 befinden sich noch Dateien anderer Formate (psd, gsp, 3ds, c4d, jvx, max, mcd, mw, wr1). Informationen und Hinweise zu diesen Dateien enthalten die entsprechenden Abschnitte der vorliegenden Arbeit bzw. Textdateien in den betreffenden Ordnern.

<sup>2</sup>POV-Ray ist unter <http://www.povray.org> frei erhältlich. Hinweise zur Beschaffung weiterer für diese Arbeit verwendeter Softwarepakete enthält die Internetseite <http://www.afiller.de/habil>.

# Literaturverzeichnis

- [1] ABEL, H. : *GPS: Global Positioning System – Funktionsweise und mathematische Grundlagen*. 2002. – [http://www.skytec.ch/Ballontechnik/doc\\_GPS.pdf](http://www.skytec.ch/Ballontechnik/doc_GPS.pdf)
- [2] ALTEN, H.-W. [u. a.] : *4000 Jahre Algebra*. Berlin: Springer, 2003
- [3] ANDRASCHKO, H. : DreiDGeo – ein Computerwerkzeug für die analytische Geometrie im  $\mathbf{E}^3$ . In: *MU* 47 (2001), 5, S. 54–68
- [4] ARTMANN, B. ; TÖRNER, G. : *Lineare Algebra und Geometrie. Grund- und Leistungskurs*. Göttingen: Vandenhoeck & Ruprecht, 1988 (2. Aufl.)
- [5] ARZT, K. (Hrsg.) ; SELINKA, M. (Hrsg.) ; STARK, J. (Hrsg.): *Grundkurs Mathematik-Abitur. Aufgaben mit Lösungen 1997–2002*. Stuttgart: Klett, 2002
- [6] AUMANN, G. ; SPITZMÜLLER, K. : *Computerorientierte Geometrie*. Mannheim: B.I. Wissenschaftsverlag, 1993
- [7] BÄR, G. : *Geometrie. Eine Einführung in die Analytische und Konstruktive Geometrie*. Stuttgart: Teubner, 1996
- [8] BAUM, M. ; LIND, D. ; SCHERMULY, H. ; WEIDIG, I. ; ZIMMERMANN, P. : *Lambacher/Schweizer. Lineare Algebra mit analytischer Geometrie. Leistungskurs*. Stuttgart: Klett, 2004
- [9] BAUMANN, R. : *Elementare Computergrafik*. Stuttgart: Klett, 1994
- [10] BAUMANN, R. : Projekte im Mathematikunterricht – geht denn das? Was der Mathematikunterricht vom Informatikunterricht lernen kann. In: *LogIn* 18 (1998), 2, S. 33–45
- [11] BENDER, M. ; BRILL, M. : *Computergrafik*. München: Hanser, 2003
- [12] BENDER, P. (Hrsg.) [u. a.] : *Proceedings 20. Lehr- und Lernprogramme für den Mathematikunterricht*. Hildesheim: Franzbecker, 2003
- [13] BENDER, P. (Hrsg.) [u. a.] : *Proceedings 21. WWW und Mathematik – Lehren und Lernen im Internet*. Hildesheim: Franzbecker, 2005

- [14] BENDER, P. ; SCHREIBER, A. : *Operative Genese der Geometrie*. Wien / Stuttgart: Hölder-Pichler-Tempsky / Teubner, 1985
- [15] BENINI, M. : Beleuchtung dreidimensionaler Körper. In: *Wurzel* (2001), 3/4, S. 50–52
- [16] BENNET, D. : *Exploring Geometry with The Geometer's Sketchpad*. Emeryville: Key Curriculum Press, 2003
- [17] BERTA, T. : Combination of traditional and computer based tools in mathematics education. In: *ZDM* 35 (2003), 1, S. 1–5
- [18] BERTUCH, M. : Polygonfabriken. Wie Grafikchips Bits und Bytes in plastische Bilder verwandeln. In: *c't* 17 (2000), 8, S. 202–207
- [19] BERTUCH, M. : Aufklärung in 3D. Was Sie schon immer über Grafikchips wissen wollten. In: *c't* 19 (2002), 15, S. 194–201
- [20] BESUDEN, H. : Raumvorstellung und Geometrieverständnis. In: *Mathematische Unterrichtspraxis (MUP)* (1999), III, S. 1–10
- [21] BEUTELSPACHER, A. : *Lineare Algebra*. Braunschweig: Vieweg, 2003 (6. Aufl.)
- [22] BEUTELSPACHER, A. ; ROSENBAUM, U. : *Projektive Geometrie*. Braunschweig: Vieweg, 2004 (2. Aufl.)
- [23] BIELIG-SCHULZ, G. ; SCHULZ, C. : *3D-Graphik in Pascal*. Stuttgart: Teubner, 1987
- [24] BIELIG-SCHULZ, G. ; SCHULZ, C. : Algorithmen zur 3D-Graphik. Eine Anwendung von analytischer Geometrie und linearer Algebra in der Informatik. In: GRAF, K. D. (Hrsg.): *Computer in der Schule; Band 2: Beispiele für Mathematikunterricht und Informatikunterricht*, Stuttgart: Teubner, 1988, S. 97–109
- [25] BIELIG-SCHULZ, G. ; SCHULZ, C. : Analytische Geometrie, lineare Optimierung und Computergraphik. In: *DdM* 16 (1988), 4, S. 304–319
- [26] BIGALKE, A. ; KÖHLER, N. : *Mathematik 11*. Düsseldorf: Cornelsen, 1988
- [27] BIGALKE, A. ; KÖHLER, N. : *Mathematik 13.1*. Düsseldorf: Cornelsen, 1990
- [28] BIRN, J. : *Lighting und Rendering*. München: Markt und Technik, 2001
- [29] BORNELEIT, P. ; DANCKWERTS, R. ; HENN, H.-W. ; WEIGAND, H.-G. : Expertise zum Mathematikunterricht in der gymnasialen Oberstufe. In: *JMD* 22 (2001), 1, S. 73–90

- [30] BORWEIN (Hrsg.) ; MORALES (Hrsg.) ; POLTHIER (Hrsg.) ; RODRIGUES (Hrsg.): *Multimedia Tools for Communicating Mathematics*. Berlin: Springer, 2001
- [31] BRÜDERLIN, B. ; MEIER, A. : *Computergrafik und Geometrisches Modellieren*. Stuttgart: Teubner, 2001
- [32] BRUGGER, R. : *3D-Computergrafik und -animation*. Bonn: Addison-Wesley, 1994
- [33] BRUGGER, R. : *Professionelle Bildgestaltung in der 3D-Computergrafik*. Bonn: Addison-Wesley, 1995
- [34] BUNGARTZ, H.-J. ; GRIEBEL, M. ; ZENGER, C. : *Einführung in die Computergraphik*. Braunschweig: Vieweg, 2002 (2. Aufl.)
- [35] BUNGARTZ, P. : Warum nur Lineare Algebra in einem Kurs „Lineare Algebra und Analytische Geometrie“? In: *BzMU 1979*, Hannover: Schroedel, 1979, S. 90–93
- [36] BÜRGER, H. ; FISCHER, R. ; MALLE, G. ; REICHEL, H.-C. : Zur Einführung des Vektorbegriffes: Arithmetische Vektoren mit geometrischer Deutung. In: *JMD* 1 (1980), 3, S. 171–187
- [37] CAMPAGNA, S. ; KOBELT, L. : Dreiecksbeziehungen. Effiziente 3D-Darstellung durch hierarchische Polygonmodelle. In: *c't* 15 (1998), 16, S. 174–179
- [38] CATTAROZZI, M. : *Color Management*. Kilchberg: Smartbooks Publishing, 2001
- [39] COHEN, M. F. ; WALLACE, J. R.: *Radiosity and realistic image synthesis*. Boston: Academic Press, 1993
- [40] COMPUTER ARTS (REDAKTION): Group Test 3D Animation Software. In: *Computer Arts* 105 (2005), 1, S. 88–93
- [41] CREUTZIG, C. ; OEVEL, W. ; GERHARD, J. : *Das MuPAD Tutorium*. Berlin: Springer, 2004 (3. Aufl.)
- [42] CUNNINGHAM, S. : Some strategies for using visualization in mathematics teaching. In: *ZDM* 26 (1994), 3, S. 83–85
- [43] CUNNINGHAM, S. (Hrsg.) ; HUBBOLD, R. (Hrsg.): *Interactive Learning Through Visualization. The Impact of Computer Graphics in Education*. Berlin: Springer, 1992
- [44] DE BERG, M. ; VAN KREVELD, M. ; OVERMARS, M. ; SCHWARZKOPF, O. : *Computational Geometry. Algorithms and Applications*. Berlin: Springer, 1997
- [45] DE LA FLOR, M. : *Carrara Studio 3 Handbook*. Hingham: Charles River, 2003

- [46] DECI, E. L. ; RYAN, R. M.: Die Selbstbestimmungstheorie der Motivation und ihre Bedeutung für die Pädagogik. In: *Zeitschrift für Pädagogik* 39 (1993), 2, S. 223–238
- [47] DIGITAL PRODUCTION (REDAKTION): Marktübersicht 3D-Software. In: *Digital Production* 7 (2003), 2, S. 130–138
- [48] DMV: Denkschrift der Deutschen Mathematiker-Vereinigung zum Mathematikunterricht an Gymnasien. In: *DdM* 5 (1977), 1, S. 70–78
- [49] DRIEMEYER, T. : *Rendering with mental ray*. Wien: Springer, 2001
- [50] DUIN, H. ; SYMANZIK, G. ; CLAUSSEN, U. : *Beleuchtungsalgorithmen in der Computergrafik*. Berlin: Springer, 1993
- [51] DÜNOW, J. : Pixelkunst. 3D-Grafik zwischen Prozessor und Monitor. In: *c't* 16 (1999), 2, S. 130–135
- [52] DZEWAS, J. (Hrsg.) ; CUKROWICZ, J. (Hrsg.): *Hahn/Dzewas. Lineare Algebra / Analytische Geometrie*. Braunschweig: Westermann, 1992
- [53] DZUNG WONG, B. : *Bézierkurven: gezeichnet und gerechnet. Ein elementarer Zugang und Anwendungen*. Zürich: Orell Füssli, 2003
- [54] EHRLER, R. : *Raytracing Lösungen*. Feldkirchen: Franzis, 1995
- [55] ELSCHENBROICH, H.-J. : Einführung in die Perspektive – Einsatz eines Dynamischen Geometrie-Systems in der Klasse 9. In: *[114]*, S. 43–48
- [56] ELSCHENBROICH, H.-J. ; MEINERS, J.-C. : *Computergraphik und Darstellende Geometrie im Unterricht der Linearen Algebra*. Bonn: Dümmler, 1994
- [57] ENCARNACAO, J. ; STRASSER, W. : *Computer Graphics*. München: Oldenbourg, 1988 (3. Aufl.)
- [58] ENDL, K. : *Analytische Geometrie und Lineare Algebra*. Düsseldorf: VDI, 1985
- [59] ENDL, K. : *Kreative Computergrafik*. Düsseldorf: VDI, 1986
- [60] ENDL, K. : *Computergrafik 1*. Biebertal-Vetzberg: Würfel-Verlag, 1989
- [61] ENDL, K. : Computergrafik im Unterricht. In: *PM* 32 (1990), 5, S. 193–198
- [62] ENDL, K. : *Computergrafik 2*. Biebertal-Vetzberg: Würfel-Verlag, 1991
- [63] Eovia Corporation: *Carrara 4 Reference Guide*. 2004. – <http://www.eovia.com>
- [64] FELLNER, W. D.: *Computergrafik*. Mannheim: B.I. Wissenschaftsverlag, 1992 (2. Aufl.)



- [65] FILLER, A. : *Euklidische und nichteuklidische Geometrie*. Mannheim: B.I. Wissenschaftsverlag, 1993
- [66] FILLER, A. : Tangenten an Kreise und Tangentialebenen an Kugeln. Ein Unterrichtsvorschlag für Leistungskurse in der Sekundarstufe II. In: *MidSch* 32 (1994), 10, S. 539–551
- [67] FILLER, A. : Analytische Geometrie. In: WALZ, G. (Hrsg.): *Lexikon der Mathematik* Bd. 1. Heidelberg: Spektrum, 2001, S. 72–75
- [68] FILLER, A. : Dreidimensionale Computergrafik und Analytische Geometrie. In: *math.did* 24 (2001), 2, S. 21–56
- [69] FILLER, A. : Didaktische Aspekte der Einbeziehung von Elementen der 3D-Computergrafik in das Stoffgebiet Analytische Geometrie. In: *[13]*, 2003, S. 81–94
- [70] FILLER, A. : 3D-Computergrafik im Stoffgebiet Analytische Geometrie – eine Zwischenbilanz. In: *BzMU 2004*, Hildesheim: Franzbecker, 2004, S. 165–168
- [71] FILLER, A. ; RIEPER, F. : 3D Computer Graphics and Analytic Geometry in Mathematics Education in Grammar Schools. In: *The Mathematics Education into the 21st Century Project, Proceedings of the International Conference „The Future of Mathematics Education“*, Ciechocinek, Poland, 2004, S. 39–44
- [72] FILLER, A. ; RIEPER, F. : *3D-Computergrafik ... und die Mathematik dahinter* (mit Beiträgen von GIEDING, M.). Karlsbad: Verlag Jutta Pohl, 2007
- [73] FILLER, A. ; WITTMANN, G. : Raumgeometrie vom ersten Tag an! Einstiege in die Analytische Geometrie. In: *MU* 50 (2004), 1, S. 90–103
- [74] FISCHER, A. : Mentale Modelle zum Vektorraumbezug. In: *math.did* 26 (2003), 2, S. 91–114
- [75] FISCHER, G. : *Lineare Algebra*. Braunschweig: Vieweg, 2003 (14. Aufl.)
- [76] FLEMING, B. : *Advanced 3D Photorealism Techniques*. New York: Wiley, 1999
- [77] FLETCHER, T. J.: Der Geometrieunterricht. Aktuelle Probleme und Zielstellungen. In: *MU* 21 (1974), 1, S. 19–35
- [78] FOLEY, J. D. ; VAN DAM, A. ; FEINER, S. K. ; HUGHES, J. F.: *Computer Graphics. Principles and Practice*. Workingham: Addison-Wesley, 1993 (2. Aufl.)
- [79] FOLEY, J. D. ; VAN DAM, A. ; FEINER, S. K. ; HUGHES, J. F. ; PHILLIPS, R. L.: *Grundlagen der Computergraphik*. Bonn: Addison-Wesley, 1994

- [80] FÖRSTER, F. (Hrsg.) ; HENN, H.-W. (Hrsg.) ; MEYER, J. (Hrsg.): *Istron. Materialien für einen realitätsbezogenen Mathematikunterricht, Band 6*. Hildesheim: Franzbecker, 2000
- [81] FRANK, B. : *Das Arbeiten mit Koordinatensystemen und Vektoren im Mathematikunterricht – eine Übersicht über historische Entwicklungen und gegenwärtige Tendenzen sowie eine Argumentation zu seiner Weiterentwicklung an der POS der DDR*, Humboldt-Universität zu Berlin, Dissertation (B), 1983
- [82] FRANK, B. ; LEMKE, H. ; STOYE, W. : *Mathematik 12*. Berlin: Volk und Wissen, 1981
- [83] FRANKE, H. W.: *Animation mit Mathematica*. Berlin: Springer, 2001
- [84] FREUDENTHAL, H. : *Mathematik als pädagogische Aufgabe*. Stuttgart: Klett, 1973
- [85] FÜHRER, L. : Objektstudien in der Vektorgeometrie. In: *DdM* 7 (1979), 1, S. 32–61
- [86] FÜHRER, L. : Über einige Grundfragen künftiger Geometriedidaktik. In: *math.did* 25 (2002), 1, S. 55–78
- [87] GALLARDO, A. : *3D Lighting: History, Concepts and Techniques*. Rockland: Charles River, 2001
- [88] GECK, E. : Der Lehrplanerlass für die höheren Knabenschulen des Königreiches Württemberg vom 27. 08. 1912. In: *Abhandlungen über den mathematischen Unterricht in Deutschland* 2 (1913), 8
- [89] GEHRS, K. : *Analytische Geometrie mit MuPAD*. Paderborn: Sciface, 2004. – <http://www.sciface.com/education/>
- [90] GIEDING, M. : *Möglichkeiten und Aspekte des Arbeitens mit Elementen der Computergrafik im Geometrieunterricht der Abiturstufe in der DDR*, Humboldt-Universität zu Berlin, Dissertation, 1988
- [91] GIEDING, M. : Wider die Armut an geometrischen Formen im Unterricht zur Linearen Algebra / Analytischen Geometrie. In: *MidSch* 30 (1992), 7, S. 407–415
- [92] GIEDING, M. : Programming by Example - Überlegungen zu den Grundlagen einer Didaktik der Tabellenkalkulation. In: *math.did* 26 (2003), 2, S. 42–72
- [93] GLAESER, G. : *Geometrie und ihre Anwendungen in Kunst, Natur und Technik*. München: Elsevier, 2005
- [94] GLASSNER, A. S. (Hrsg.): *An Introduction to Ray Tracing*. London: Academic Press, 1989

- [95] GRAUMANN, G. ; HÖLZL, R. ; NEUBRAND, M. ; STRUVE, H. : Tendenzen der Geometriedidaktik der letzten 20 Jahre. In: *JMD* 17 (1996), 3/4, S. 163–237
- [96] GRAY, A. : *Differentialgeometrie*. Heidelberg: Spektrum, 1994
- [97] GRIESEL, H. (Hrsg.) ; POSTEL, H. (Hrsg.): *Mathematik heute. Leistungskurs Lineare Algebra / Analytische Geometrie*. Hannover: Schroedel-Schöningh, 1986
- [98] GUDERIAN, D. : Mathematik und Kunst. In: *ZDM* 22 (1990), 2, S. 60–67
- [99] GUTZMER, A. : *Die Tätigkeit der Unterrichtskommission der Gesellschaft Deutscher Naturforscher und Ärzte*. Leipzig: Teubner, 1908
- [100] GUTZMER, A. : Bericht betreffend den Unterricht an den neunklassigen höheren Lehranstalten (Reformvorschläge von Meran). In: *MU* 27 (1980), 6, S. 53–62
- [101] HABERSTROH, T. : Grundlagen der 3D-Computergrafik. In: *LogIn* 15 (1995), 2, S. 15–20
- [102] HAENSELMANN, T. : *Raytracing*. Bonn: Addison-Wesley, 1996
- [103] HAFENBRAK, B. : Bericht zur Arbeitsgruppe „Computergrafik als Bindeglied zwischen Mathematik- und Informatikunterricht“. In: *[113]*, S. 178–179
- [104] HANDKE, J. : *Training Lineare Algebra und Analytische Geometrie*. Stuttgart: Klett, 2004
- [105] HASE, H.-L. : *Dynamische virtuelle Welten mit VRML 2.0*. Heidelberg: dpunkt, 1997
- [106] HÄSSLER, U. : *3D Imaging*. Berlin: Springer, 1997
- [107] HAUBROCK, D. : GPS in der analytischen Geometrie. In: *[80]*, S. 86–103
- [108] HEISS, P. ; HERMES, A. ; BAUMANN, R. (Hrsg.) ; HERMES, A. (Hrsg.) ; REIMER, R. (Hrsg.): *3D-Grafik (Arbeitshefte Informatik)*. Stuttgart: Klett, 1995
- [109] HENN, H.-W. : Computer-Algebra Systeme – junger Wein oder neue Schläuche? In: *JMD* 25 (2004), 3/4, S. 73–90
- [110] HENN, H.-W. (Hrsg.) ; MAASS, K. (Hrsg.): *Istron. Materialien für einen realitätsbezogenen Mathematikunterricht, Band 8*. Hildesheim: Franzbecker, 2003
- [111] HERGET, W. (Hrsg.) [u. a.] : *Proceedings 17. Standardthemen des Mathematikunterrichts in moderner Sicht*. Hildesheim: Franzbecker, 2000
- [112] HISCHER, H. ; WEIGAND, H.-G. : Mathematikunterricht und Informatik – Gedanken zur Veränderung eines Unterrichtsfachs. In: *LogIn* 18 (1998), 2, S. 10–18

- 
- [113] HISCHE, H. (Hrsg.): *Proceedings 14. Computer und Geometrie*. Hildesheim: Franzbecker, 1997
- [114] HISCHE, H. (Hrsg.): *Proceedings 15. Geometrie und Computer – Suchen, Entdecken und Anwenden*. Hildesheim: Franzbecker, 1998
- [115] HISCHE, H. (Hrsg.): *Proceedings 16. Modellbildung, Computer und Mathematikunterricht*. Hildesheim: Franzbecker, 2000
- [116] HISCHE, H. : *Mathematikunterricht und Neue Medien*. Hildesheim: Franzbecker, 2002
- [117] HISCHE, H. ; SCHEID, H. : *Grundbegriffe der Analysis. Genese und Beispiele aus didaktischer Sicht*. Heidelberg: Spektrum, 1995
- [118] VOM HOFE, R. : *Grundvorstellungen mathematischer Inhalte*. Heidelberg: Spektrum, 1995
- [119] HORMANN, C. : Mathe schafft Natur. Algorithmische 3D-Landschaften in POV-Ray. In: *c't* 20 (2003), 20, S. 206–211
- [120] HUSSMANN, S. : *Mathematik entdecken und erforschen. Theorie und Praxis des Selbstlernens in der Sekundarstufe II*. Berlin: Cornelsen, 2003
- [121] INHETVEEN, H. : *Die Reform des gymnasialen Mathematikunterrichts zwischen 1890 und 1914*. Bad Heilbrunn: Verlag Julius Klinkhardt, 1976
- [122] JÄHNE, B. : *Digitale Bildverarbeitung*. Berlin: Springer, 2002 (5. Aufl.)
- [123] JAHNKE, T. : *Historische und didaktische Untersuchungen zum Aufbau der Differentialrechnung*. Bad Salzdetfurth: Franzbecker, 1989
- [124] JANSER, A. ; LUTHER, W. ; OTTEN, W. : *Computergraphik und Bildverarbeitung*. Braunschweig: Vieweg, 1996
- [125] JONES, H. : *Computer Graphics through Key Mathematics*. London: Springer, 2001
- [126] KADUNZ, G. : Visualisierung, Bild und Metapher. Die vermittelnde Tätigkeit der Visualisierung beim Lernen von Mathematik. In: *JMD* 21 (2000), 3/4, S. 280–302
- [127] KARSAI, J. ; RACZ, E. V. ; SCHWENK, A. ; KALUS, N. : Visualization and art in the mathematics classroom. In: *ZDM* 35 (2003), 1, S. 24–29
- [128] KAUSCHITSCH, H. (Hrsg.) ; METZLER, W. (Hrsg.): *Medien zur Veranschaulichung von Mathematik*. Wien / Stuttgart: Hölder-Pichler-Tempsky / Teubner, 1987

- 
- [129] KAUTSCHITSCH, H. : Bilder im Mathematikunterricht. In: *MU* 33 (1987), 4, S. 37–48
- [130] KAUTSCHITSCH, H. : „Neue“ Anschaulichkeit durch „neue“ Medien. In: *ZDM* 26 (1994), 3, S. 79–82
- [131] KLEIFELD, A. : Geometrisches Modellieren mit Bézierkurven verbindet anwendungsbezogen Analysis, Lineare Algebra und Algorithmik. In: *[80]*, S. 61–79
- [132] KLEIFELD, A. : Verformbare Kurven. Geometrisches Modellieren mit Bézierkurven. In: *[115]*, S. 120–124
- [133] KLEMENZ, H. : Ein plattformunabhängiges Computerwerkzeug für die dynamische Raumgeometrie. In: *MU* 47 (2001), 5, S. 69–77
- [134] KLIKA, M. : Zentrale Ideen – echte Hilfen. In: *ml* 119 (2003), S. 4–7
- [135] KLINCSIK, M. : Teaching spline approximation techniques using Maple. In: *ZDM* 35 (2003), 2, S. 30–35
- [136] KMK: *Einheitliche Prüfungsanforderungen im Fach Mathematik. Beschluss der 298. Kultusministerkonferenz am 23./24.05.2002 in Eisenach.* Berlin, 2002
- [137] KNECHTEL, H. : Die Behandlung von Spline-Funktionen im Analysisunterricht. In: *[80]*, S. 25–38
- [138] KÖBERLE, J. : *Computerunterstützter Einstieg in die Analytische Geometrie*, Landeslehrerprüfungsamt Baden-Württemberg, Außenstelle Karlsruhe, Schriftliche Prüfungsarbeit für die zweite Staatsprüfung für die Laufbahn des höheren Schuldienstes an Gymnasien, 2005
- [139] KOECHER, M. : *Lineare Algebra und analytische Geometrie.* Berlin: Springer, 1999 (4. Aufl.)
- [140] KOFLER, M. ; GRÄBE, H.-G. : *Mathematica.* Bonn: Addison-Wesley, 2002 (4. Aufl.)
- [141] KROLL, W. : Ebene Kurven im Raum – ein weites Feld für die sinnreiche Verbindung von Analysis und Analytischer Geometrie. In: *MNU* 51 (1998), 7, S. 420–425
- [142] KRÖMER, R. : *Zur Geschichte des axiomatischen Vektorraumbegriffs*, Universität des Saarlandes, Saarbrücken, Diplomarbeit, 1998
- [143] KRUMPE, N. : Creating Three-Dimensional Scenes. In: *Mathematics Teacher* 98 (2005), 6, S. 394–398

- [144] KUCKWA, U. : Einführung in die Computergrafik. Ein Unterrichtsprojekt. In: *LogIn* 15 (1995), 2, S. 31–37
- [145] KUNATH, L. ; UHLMANN, E. : Wie kann man Probleme der Vektorrechnung und der analytischen Geometrie in Klasse 12 veranschaulichen? In: *MidSch* 19 (1981), 7-8, S. 570–580
- [146] KUSKA, J. P.: Frei im Raum. Mathematica 3D-Grafik in Bewegung. In: *c't* 14 (1997), 9, S. 292–298
- [147] LAMA, T. : *3D-Welten*. München: Hanser, 2004
- [148] LANGTANGEN, H. P.: *Python Scripting for Computational Science*. Berlin: Springer, 2004
- [149] LEHMANN, E. : Abbildungsgeometrie mit Matrizen als Grundlage der Computergrafik. In: *[113]*, S. 134–141
- [150] LEHMANN, E. : CAS-Bausteine bei der Modellierung mathematischer Standardthemen. In: *[111]*, S. 25–31
- [151] LEHMANN, E. : Impressionen an Kreisen. In: *[114]*, S. 69–73
- [152] LEHMANN, E. : Lineare Algebra und Analytische Geometrie. Ein Kurskonzept auf den Grundlagen von Matrizenrechnung und Computereinsatz. In: *MU* 39 (1993), 4, S. 31–64
- [153] LEHMANN, E. : *Konzeptionelle Überlegungen zur Einbeziehung informatischer Inhalte und Methoden beim Computereinsatz im Mathematikunterricht der Sekundarstufe 2*. Hildesheim: Franzbecker, 2004
- [154] LENNÉ, H. : *Analyse der Mathematikdidaktik in Deutschland*. Stuttgart: Klett, 1969
- [155] LEPPIG, M. : Visualisierung und intuitives Arbeiten. In: *MU* 34 (1988), 5, S. 21–25
- [156] LEUDERS, T. : Raumgeometrie mit dem Computer – Schülerprojekte in 2 bis 5 Raumdimensionen. In: *[12]*, S. 105–111
- [157] LEUDERS, T. : *Mathematik Didaktik*. Berlin: Cornelsen Scriptor, 2003
- [158] LEUDERS, T. : Vom räumlichen Sehen zu Projektionen. Die kartesische Idee (nach-) erfinden. In: *ml* 119 (2003), S. 52–56
- [159] LEUDERS, T. : Kreatives geometrisches Konstruieren mit Ray-Tracing Software. In: LEUDERS, T. (Hrsg.): *Materialien für einen projektorientierten Mathematik- und Informatikunterricht*, Hildesheim: Franzbecker, 2004, S. 17–26

- [160] LIETZMANN, W. : *Stoff und Methode im mathematischen Unterricht der nord-deutschen höheren Schulen*. Leipzig: Teubner, 1909
- [161] LIND, D. : *Koordinaten, Vektoren, Matrizen*. Heidelberg: Spektrum, 1997
- [162] LINDNER, W. : CAS-supported multiple representations in elementary linear algebra. The case of the Gaussian algorithm. In: *ZDM* 35 (2003), 2, S. 36–42
- [163] LINDNER, W. : *Lineare Transformationen mit MuPAD*. Paderborn: Sciface, 2004.  
– <http://www.sciface.com/education/>
- [164] LOVISCACH, J. : Es werde Licht. Verfahren zur 3D-Darstellung. In: *c't* 14 (1997), 1, S. 198–204
- [165] LOVISCACH, J. : Digitale Legenden. 3D-Software der High-End-Klasse. In: *c't* 18 (2001), 11, S. 140–153
- [166] LOVISCACH, J. : Pixel-Explosionen. 3D-Softwarepakete der Mittelklasse. In: *c't* 18 (2001), 12, S. 164–177
- [167] LOVISCACH, J. : Siggraph 2005: 3D-Software, Echtzeit-Raytracing. In: *c't* 22 (2005), 18, S. 60–62
- [168] LOWTHER, J. L. ; SHENE, C.-K. : Education: Rendering + Modeling + Animation + Postprocessing = Computer Graphics. In: *Computer Graphics* 34 (2000), 4, S. 15–18
- [169] LUTHER, W. ; OHSMANN, M. : *Mathematische Grundlagen der Computergrafik*. Braunschweig: Vieweg, 1989 (2. Aufl.)
- [170] MAASS, K. : Sicher durch die Lüfte - Geraden und Ebenen, die sich nicht schneiden dürfen. In: *[110]*, S. 178–202
- [171] MAEDER, R. E.: Ray Tracing and Graphics Extensions. In: *The Mathematica Journal* 4 (1994), 3
- [172] MAJEWSKI, M. : Ray tracing in the classroom. In: *International journal of mathematical education in science and technology* 28 (1997), 2, S. 211–223
- [173] MAJEWSKI, M. : *MuPAD Pro Computing Essentials*. Berlin: Springer, 2004 (2. Aufl.)
- [174] MAJEWSKI, M. : Reflections on the Use of Computer Graphics in Teaching Mathematics. In: *mathPAD* 14 (2005), 1. – <http://www.mupad.com/mathpad>
- [175] MARCHESE, F. T.: Teaching computer graphics with spreadsheets. In: GRISSON, S. (Hrsg.) [u. a.] : *Conference abstracts and applications: SIGGRAPH 98*, New York: ACM Press, 1998, S. 84–87

- 
- [176] MARTELLI, A. : *Python in a Nutshell*. Sebastopol: O'Reilly, 2003
- [177] MEALING, S. : *The Art and Science of Computer Animation*. Exeter: Intellect Books, 1998
- [178] MEYER, J. : Bahnkurven als geometrische Objekte – Vernetzung durch Variation. In: [113], S. 90–95
- [179] MEYER, J. : Bézierkurven. In: [80], S. 44–60
- [180] MEYER, J. : Bézierkurven als Modellierung von Designerkurven. In: [115], S. 125–130
- [181] MEYER, J. : Eine Fläche mit einer Falte. In: [114], S. 89–94
- [182] MEYER, J. : Projektionen. In: [80], S. 104–117
- [183] MEYER, J. : Von der Normalparabel zu kubischen Kurven. In: *math.did* 21 (1998), 2, S. 84–108
- [184] MEYER, J. : Kurven und Flächen in der Vektorgeometrie. In: *math.did* 24 (2001), 1, S. 51–70
- [185] MÜLLER, J. : Realistische Darstellungen; Teil 1: Grundlagen; Teil 2: Raytracing; Teil 3: Animation. In: *LogIn* 15 (1995), 2, 4, 5/6, S. 38–44, 40–45, 60–64
- [186] NEUE LEHRPLÄNE FÜR DEN MATHEMATISCHEN UND NATURWISSENSCHAFTLICHEN UNTERRICHT AN DEN HÖHEREN LEHRANSTALTEN: Nach den Lehrplänen vom Jahre 1905 neubearbeitet vom Deutschen Ausschuss für den mathematischen und naturwissenschaftlichen Unterricht (1922). In: *MU* 27 (1980), 6, S. 63–80
- [187] NEUMANN, B. : *Bildverarbeitung für Einsteiger*. Berlin: Springer, 2004
- [188] NICOL, M. : *Potenzen des Arbeitens mit Kurven und deren Gleichungen hinsichtlich einer Weiterentwicklung des Mathematikunterrichts der Abiturstufe in der DDR*, Humboldt-Universität zu Berlin, Dissertation, 1987
- [189] NISCHWITZ, A. ; HABERÄCKER, P. : *Masterkurs Computergrafik und Bildverarbeitung*. Braunschweig: Vieweg, 2004
- [190] NYMAN, M. : *4 Farben – 1 Bild*. Berlin: Springer, 1999 (3. Aufl.)
- [191] OLDENBURG, R. : Schüler begegnen der Mathematik der dritten Dimension. In: *DMV-Mitteilungen* 12 (2004), 2, S. 82–83
- [192] OLDENBURG, R. : Von XRay bis XLab: Computertomographie im MU. In: *BzMU 2004*, Hildesheim: Franzbecker, 2004, S. 421–424



- [193] OLDENBURG, R. : Rekonstruktion von 3D-Koordinaten aus Bildern. In: *Istron 9*, Hildesheim: Franzbecker, erscheint 2006
- [194] OLDENBURG, R. : Mathematische Schülerexperimente. In: *ml* (erscheint 2007)
- [195] ORLAMÜNDER, D. ; MASCOLUS, W. : *Computergrafik und OpenGL*. München: Hanser, 2004
- [196] OWEN, G. S. ; LARRONDO-PETRIE, M. M. ; LAXER, C. : Computer Graphics Curriculum: Time for a change? In: *Computer Graphics* 28 (1994), 3, S. 183–185
- [197] PAREIGIS, B. : *Analytische und projektive Geometrie für die Computer-Graphik*. Stuttgart: Teubner, 1990
- [198] PEREZ, J.-P. : *Optik*. Heidelberg: Spektrum, 1996
- [199] POLYA, G. : *Schule des Denkens*. Bern: Francke, 1949
- [200] POV-Team: *POV-Ray Reference*. 2004. – <http://www.povray.org/documentation>
- [201] PRATT, W. K.: *Digital Image Processing*. New York: Wiley, 2001 (3. Aufl.)
- [202] PRESMEG, N. C.: The role of visually mediated processes in classroom mathematics. In: *ZDM* 26 (1994), 4, S. 114–117
- [203] PROFKE, L. : Zur Behandlung der linearen Algebra in der Sekundarstufe II. In: *BzMU 1977*, Hannover: Schroedel, 1977, S. 207–210
- [204] PROFKE, L. : Zur Behandlung der linearen Algebra in der Sekundarstufe II. In: WINKELMANN, B. (Hrsg.): *Materialien zur Linearen Algebra und Analytischen Geometrie in der Sekundarstufe II. IDM Materialien und Studien. Band 13*, Universität Bielefeld, 1978, S. 10–42
- [205] PROFKE, L. : Kegelschnitte im Mathematikunterricht? In: *MidSch* 30 (1992), 11, S. 603–615
- [206] PROFKE, L. : Kegelschnitte – ein Lehrgang (Teile 1 und 2). In: *MidSch* 31 (1993), 1, 2, S. 19–28, 112–117
- [207] RAAKE, R. : *Mathcad in der Schule*. Berlin: Paetec, 2002
- [208] REICHEL, H.-C. : Zum Skalarprodukt in der Sekundarstufe, eine didaktische Analyse. In: *DdM* 8 (1980), 2, S. 102–132
- [209] REICHEL, H.-C. (Hrsg.): *Computereinsatz im Mathematikunterricht*. Mannheim: B.I. Wissenschaftsverlag, 1995
- [210] RICHTER, K. : *Computergrafik und Farbmetrik*. Berlin: VDE-Verlag, 1994

- 
- [211] ROCK, I. : *Wahrnehmung. Vom visuellen Reiz zum Sehen und Erkennen*. Heidelberg: Spektrum, 1985
- [212] ROOSENDAAL, T. ; WARTMANN, C. : *Der offizielle Blender 2.0 Guide*. München: Markt und Technik, 2001
- [213] ROTH, N. : Ein integratives Grundkurskonzept am Beispiel der Bézierkurven. In: *math.did* 25 (2002), 2, S. 95–113
- [214] SCHAPER, R. : Computergraphik und Visualisierung am Beispiel zweier Themen aus der linearen Algebra. In: *ZDM* 26 (1994), 3, S. 86–92
- [215] SCHLEGEL, K. : *Vom Regenbogen zum Polarlicht. Leuchterscheinungen in der Atmosphäre*. Heidelberg: Spektrum, 1995
- [216] SCHMID, A. (Hrsg.) ; SCHWEIZER, W. (Hrsg.): *Lambacher / Schweizer. Analytische Geometrie mit Linearer Algebra. Leistungskurs*. Stuttgart: Klett, 1988
- [217] SCHMID, A. (Hrsg.) ; SCHWEIZER, W. (Hrsg.): *Lambacher / Schweizer. Analytische Geometrie mit Linearer Algebra. Grundkurs*. Stuttgart: Klett, 1990
- [218] SCHMIDT, G. : Curriculare Gedanken und Reflexionen zur Analytischen Geometrie (und Linearen Algebra) im Unterricht der gymnasialen Oberstufe. In: *MU* 39 (1993), 4, S. 15–30
- [219] SCHNEIDER, P. J. ; EBERLY, D. H.: *Geometric Tools for Computer Graphics*. San Francisco: Morgan Kaufmann Publishers, 2002
- [220] SCHÖNHERR, M. : Welten-Bau. Wie 3D-Programme Objekte modellieren. In: *c't* 21 (2004), 2, S. 196–201
- [221] SCHREIBER, P. : Die Koordinatenmethode im Wandel der Zeiten. Zum 350. Jahrestag des Erscheinens von R. Descartes „La Geometrie“. In: *Alpha. Mathematische Schülerzeitschrift* 21, 22 (1987, 1988), 6, 1, S. 122–123, 6–7
- [222] SCHRÖDER, E. : *Kartenentwürfe der Erde*. Leipzig: Teubner, 1988
- [223] SCHULZ, W. (Hrsg.) ; STOYE, W. (Hrsg.): *Analytische Geometrie, Grundkurs*. Berlin: Volk und Wissen, 1998
- [224] SCHULZ, W. (Hrsg.) ; STOYE, W. (Hrsg.): *Analytische Geometrie, Leistungskurs*. Berlin: Volk und Wissen, 1998
- [225] SCHUMANN, H. : Computerunterstütztes Erzeugen und Darstellen von Körperschnitten - Ein Beitrag zum allgemeinbildenden Geometrieunterricht, Teil 1. In: *DdM* 22 (1994), 4, S. 283–308

- [226] SCHUMANN, H. : New Standards for the Solution of Geometric Calculation Problems by Using Computers. In: *ZDM* 29 (1997), 5, S. 155–161
- [227] SCHUMANN, H. : Computerunterstütztes Lösen offener raumgeometrischer Aufgaben. In: *ZDM* 32 (2000), 6, S. 175–185
- [228] SCHUMANN, H. : *Raumgeometrie – Unterricht mit Computerwerkzeugen*. Berlin: Cornelsen, 2001
- [229] SCHUMANN, H. : Unterrichtssoftware für die Raumgeometrie. In: *MU* 47 (2001), 5, S. 6–10
- [230] SCHUMANN, H. : Computer aided treatment of 3d-problems in analytic geometry. In: *ZDM* 35 (2003), 1, S. 7–13
- [231] SCHUPP, H. : Regeometrisierung der Schulgeometrie – durch Computer? In: *[113]*, S. 16–25
- [232] SCHUPP, H. : Mit oder ohne Computer? Computergraphische Hilfen beim Konstruieren von Kegelschnitten. In: *ml* 37 (1989), S. 49–52
- [233] SCHUPP, H. : Mathematische Phänomene rund um die Gleitstrecke. In: *DdM* 19 (1991), 1, S. 27–39
- [234] SCHUPP, H. : Geometrie in der Sekundarstufe II. In: *JMD* 21 (2000), 1, S. 50–60
- [235] SCHUPP, H. : *Kegelschnitte*. Hildesheim: Franzbecker, 2000
- [236] SCHWILL, A. : Graphische Datenverarbeitung. Eine fruchtbare Verbindung von Mathematik und Informatik. In: *[113]*, S. 172–177
- [237] SCRIBA, C. J. ; SCHREIBER, P. : *5000 Jahre Geometrie*. Berlin: Springer, 2001
- [238] STECK, R. : Konstruktive Hilfe. CAD-Programme bis 760 Euro. In: *c't* 20 (2003), 4, S. 142–153
- [239] STELLER, E. : *Computer und Kunst*. Mannheim: B.I. Wissenschaftsverlag, 1992
- [240] STEPHENSON, I. : *Essential RenderMan fast*. Berlin: Springer, 2003
- [241] STRÄSSER, R. : Computerunterstützte Geometrie im Beruf: CAD? Oder mehr? Oder anders? In: *[114]*, S. 25–33
- [242] TANIMOTO, S. L.: Mathematics education using computer graphics. In: *Computer Graphics* 25 (1991), 3, S. 165–167
- [243] TIETZE, U.-P. : Die Bedeutung lehr- und lerntheoretischer Konzepte in der analytischen Geometrie und linearen Algebra – dargelegt an einem Teilgebiet. In: *BzMU* 1978, Hannover: Schroedel, 1978, S. 280–283

- [244] TIETZE, U.-P. : Fundamentale Ideen der linearen Algebra und analytischen Geometrie. Aspekte der Curriculumsentwicklung im MU der SII. In: *math.did* 2 (1979), 3, S. 137–164
- [245] TIETZE, U.-P. : Analytische Geometrie und Lineare Algebra im MU – Unterschiedliche Ansätze und deren didaktische Rechtfertigung. In: *math.did* 4 (1981), Sonderheft 1, S. 57–99
- [246] TIETZE, U.-P. : Fachdidaktische Grundfragen des Mathematikunterrichts in der Sekundarstufe II (unter Mitarbeit von FÖRSTER, F.). In: TIETZE (Hrsg.) ; KLIKA (Hrsg.) ; WOLPERS (Hrsg.): *Mathematikunterricht in der Sekundarstufe II* Bd. 1. Braunschweig: Vieweg, 1997
- [247] TIETZE, U.-P. : Didaktik der Analytischen Geometrie und Linearen Algebra (unter Mitarbeit von SCHROTH, P.; WITTMANN, G.). In: TIETZE (Hrsg.) ; KLIKA (Hrsg.) ; WOLPERS (Hrsg.): *Mathematikunterricht in der Sekundarstufe II* Bd. 2. Braunschweig: Vieweg, 2000
- [248] TÖNNIES, K. D. ; LEMKE, H. U.: *3D-Computergrafische Darstellungen*. München: Oldenbourg, 1994
- [249] TÖRNER, G. : Erfahrungen und Bemerkungen zu Kursen in linearer Algebra. In: *MNU* 35 (1982), 6, S. 321–325
- [250] VELSZ, I. : *3ds max 4. Grundlagen und Praxis der 3D-Visualisierung und -Animation*. München: Addison-Wesley, 2002
- [251] VINCE, J. : *3-D Computer Animation*. Workingham: Addison-Wesley, 1993 (2. Aufl.)
- [252] VOLKERT, K. : *Geschichte der Analysis*. Mannheim: B.I. Wissenschaftsverlag, 1988
- [253] VOLKERT, K. T.: *Die Krise der Anschauung*. Göttingen: Vandenhoeck & Ruprecht, 1986
- [254] VOLLRATH, H.-J. : Funktionales Denken. In: *JMD* 10 (1989), 1, S. 3–37
- [255] VOSSLER, D. L.: *Exploring Analytic Geometry with Mathematica*. London: Academic Press, 2000
- [256] WATKINS, C. D. ; SHARP, L. : *Programmieren in 3D*. Hannover: Heise, 1993
- [257] WATT, A. : *3D-Computergrafik*. München: Pearson Education, 2002
- [258] WATT, A. ; POLICARPO, F. : *The Computer Image*. Harlow: Addison-Wesley, 1998

- [259] WEBER, F. : Schulung des räumlichen Vorstellungsvermögens in der Analytischen Geometrie durch Geovek. In: *ml* 34 (1989), S. 40–44
- [260] WEBER, K. (Hrsg.) ; ZILLMER, W. (Hrsg.): *Mathematik Grundkurs Sekundarstufe II*. Berlin: Paetec, 1995
- [261] WEBER, K. (Hrsg.) ; ZILLMER, W. (Hrsg.): *TCP2001 Mathematik. Gymnasiale Oberstufe - Leistungskurs*. Berlin: Paetec, 2001
- [262] WEIGAND, H.-G. : Veränderungen des Mathematikunterrichts aufgrund des Einflusses der Informatik. In: *[114]*, S. 131–136
- [263] WEIGAND, H.-G. : Überlegungen zum Verhältnis von Mathematik- und Informatikunterricht. In: *MNU* 46 (1993), 7, S. 428–432
- [264] WEIGAND, H.-G. ; FLACHSMEYER, J. : Ein computerunterstützter Zugang zu Funktionen von zwei Veränderlichen. In: *math.did* 20 (1997), 2, S. 3–23
- [265] WEIGAND, H.-G. ; WETH, T. : *Computer im Mathematikunterricht*. Heidelberg: Spektrum, 2002
- [266] WELLER, H. : Analytische Geometrie des Raumes mit Derive – Platonische und Archimedische Körper als roter Faden. In: *[113]*, S. 142–151
- [267] WELLS, D. ; YOUNG, C. ; FARMER, D. : *Ray Tracing Creations*. Corte Madera: The Waite Group Press, 1993
- [268] WETH, T. : Was bringt der Computer „wirklich“ Neues für den Geometrieunterricht? In: *[114]*, S. 13–21
- [269] WETH, T. : *Zum Verständnis des Kurvenbegriffs im Mathematikunterricht*. Hildesheim: Franzbecker, 1993
- [270] WETH, T. : Computerunterstützung offener Aufgabenstellungen im Geometrieunterricht. In: *ZDM* 32 (2000), 6, S. 166–174
- [271] WILLIM, B. : *Leitfaden der Computer Grafik*. Berlin: Drei-R-Verlag, 1989
- [272] WINDELBERG, D. : 3D-Modellierung in der Technik. In: *[114]*, S. 34–39
- [273] WINTER, H. : *Entdeckendes Lernen im Mathematikunterricht*. Braunschweig: Vieweg, 1991 (2. Aufl.)
- [274] WINTER, H. : Mathematikunterricht und Allgemeinbildung. In: *Mitteilungen der Gesellschaft für Didaktik der Mathematik* 61 (1995), S. 37–46
- [275] WINTER, M. : Läßt sich allgemeinbildender Mathematikunterricht im Grundkurs realisieren? In: *ml* 33 (1989), S. 43–49

- [276] WITTMANN, G. : *Schülerkonzepte zur Analytischen Geometrie*. Hildesheim: Franzbecker, 2003
- [277] WITTMANN, G. : Zentrale Ideen der Analytischen Geometrie. In: *ml* 119 (2003), S. 47–51
- [278] WOLFE, R. : A Syllabus Survey: Examining the State of Current Practice in Introductory Computer Graphics Courses. In: *Computer Graphics* 33 (1999), 1, S. 29–32
- [279] WUNDERLING, H. : Was entstehen kann, wenn Lineare Algebra und Analysis in Klassenstufe 11 nicht getrennt unterrichtet werden. In: *[114]*, S. 82–88
- [280] WUNDERLING, H. : Analytische Geometrie und Lineare Algebra mit Computergraphik und Matrizenrechner. In: *ml* 34 (1989), S. 45–49
- [281] XIANG, Z. : A Nontraditional Computer Graphics Course for Computer Science Students. In: *Computer Graphics* 28 (1994), 3, S. 186–188
- [282] XIANG, Z. ; PLASTOCK, R. A.: *Computergrafik*. Bonn: mitp, 2003
- [283] ZEITLER, H. : Der Tod der Geometrie. In: *ZDM* 13 (1981), 1, S. 9–12

### Verwendete Abkürzungen für Zeitschriften und Sammelbände

- BzMU* – Beiträge zum Mathematikunterricht
- c't* – c't (Magazin für Computertechnik)
- DdM* – Didaktik der Mathematik
- JMD* – Journal für Mathematik-Didaktik
- LogIn* – LOG IN – Informatische Bildung und Computer in der Schule
- math.did* – mathematica didactica
- MidSch* – Mathematik in der Schule
- ml* – Mathematik lehren
- MNU* – Der mathematische und naturwissenschaftliche Unterricht
- MU* – Der Mathematikunterricht
- PM* – Praxis der Mathematik
- ZDM* – Zentralblatt für Didaktik der Mathematik

### Internetadressen

Die in der Arbeit angegebenen Internetadressen wurden zuletzt im Mai 2007 überprüft.

# Stichwortverzeichnis

## 3

3D-Grafiksoftware ... *siehe* Grafiksoftware  
3ds max ..... 82, 88, 144, 168, 170, 175

## A

Abbildungen

– affine ..... 113–115, 257–260  
– geometrische ..... 7–8  
– Projektionen ..... 118–119, 256–257  
– Texturprojektion ..... 124–127

ABEL, H. .... 49

Adobe Illustrator ..... 70

Adobe Photoshop ..... 68, 76

Äquivalenzklasse ..... 192

Algebraisch-axiomatische Vorgehensweisen  
im MU ..... 18–21

Algebraisierung der Geometrie ..... 9

algorithmisches Arbeiten ..... 57, 290, 291

Aliasing ..... 73, 149–150, 159

Allgemeinbildung und Analytische Geometrie  
..... 32

**ambient** ..... 134

ambiente Beleuchtung ..... 131

Analogiebetrachtungen

– Kurven-Flächen ..... 265, 269  
– Raum-Ebene ..... 38, 220

**angle** ..... *siehe* Kamera

Animationen ..... 151–159

– Behandlung im Unterricht ..... 249–254,  
259–260

– Echtzeit-Animationen ..... 158–159

– Farbanimationen ..... 155

– Formänderungen ..... 155

– geometrische Transformationen 153–154,

259–260

– Interaktivität ..... 158–159

– Kameraanimationen .. 153, 250–251, 253

– Keyframe-Animationen ..... 156

– Parameterdarstellungen ..... 151–153,  
249–254

– Pfadanimationen ..... 156

– Scheinwerferanimation ..... 153, 320

anisotropische Shader ..... 135

Anschauungsvermögen 13, 24, 40, 187, 217

Ansichtskoordinatensystem ..... 116

Anti-Aliasing ..... 73, 149–150, 159

– adaptives ..... 150

Anwendungsorientierung des MU . 13, 23,  
33, 49, 297

API ..... 160

Arbeitskreis Mathematikunterricht und In-  
formatik der GDM ..... 183

Arbeitsschritte der 3D-Computergrafik 83,  
84

Arbeitsunterricht ..... 54

Archimedes Geo3D ..... 178, 204, 205

Archimedische Spirale ..... 152, 251

Arithmetisierungsprogramm ..... 40

Array ..... 194

ARTMANN, B. .... 192

Aspekte

– der Einbeziehung von Elementen der Com-  
putergrafik in den MU ..... 189

ATHEN, H. .... 16

Aufgabendidaktik ..... 54

Aufgabeninseln ..... 15, 17, 34, 54

Auflösung von Geräten ..... 71

Ausdehnungsgebilde ..... 5

- Autograph ..... 173  
 Axiomatik ..... 5–6  
 – im Mathematikunterricht ..... 18–21
- B**
- B-Splinekurven ..... 100  
 Backface Culling ..... 122  
**background** ..... *siehe* Hintergrundfarbe  
 Bahngeschwindigkeit ..... 254  
 BAUMANN, R. .... 59  
 BAUR, A. .... 16  
 Beleuchtungsichte ..... 131  
 Beleuchtungskomponenten ..... 131–135  
 – ambiente Beleuchtung ..... 131  
 – Behandlung im Unterricht ..... 283–286  
 – diffuse Reflexion ..... 131  
 – PHONGsche Komponente ..... 133  
 – spekulare Highlights ..... 132  
 – Zusammenfassung ..... 134  
 Beleuchtungsmodelle ..... 50, 131–135  
 – Behandlung im Unterricht ..... 283–286  
 – globale ..... 145–148  
 – LAMBERTsches Modell ..... 134  
 – PHONGsches Modell ..... 134  
 BENDER, P. .... 52  
 bereichsspezifische Strategien ..... 36  
 BERNOULLI, J. .... 3  
 Bernsteinpolynome ..... 99, 107  
 Beugung ..... 74  
 Bewegungen  
 – Matrizendarstellung ..... 114  
 Bewegungsbahnen ..... 151–154  
 – Behandlung im Unterricht ..... 249–254  
 Bézierflächen ..... 106  
 Bézierkurven ..... 47, 70, 98, 289  
 Bézierpatch ..... 107  
**bicubic\_patch** ..... *siehe* Bézierpatch  
 BIELIG-SCHULZ, G. .... 181  
 Bildbearbeitung ..... 68–69  
 Bilderkennung ..... 69  
 Bildmatrix ..... 67  
 Bildraum-Algorithmen ..... 123  
 Bildschirmkoordinaten ..... 72  
 Bildungsstandards ..... 36  
 – für Baden-Württemberg ..... 30  
 Blender ..... 170  
 Bogenlänge ..... 254  
**boolean** ..... 194  
 Boolesche Operationen ..... 89  
 BORNELEIT, P. .... 33  
 BOURBAKI, N. (Gruppe BOURBAKI) .. 18  
**box** ..... *siehe* Quader  
 Brechung ..... 74, 129  
 – Wellenlängenabhängigkeit ..... 144  
 Brechungsgesetz ..... 129  
 Brechungsindex ..... 129, 130  
 BRESENHAM-Algorithmus ..... 72  
 Brightness ..... *siehe* Helligkeit  
 BÜRGER, H. .... 192, 200  
 Bump Mapping ..... 135
- C**
- Cabri 3D ..... 177–178  
 CAD (Computer Aided Design) ... 71, 171  
**camera** ..... *siehe* Kamera  
 Carrara ..... 166, 170, 175  
 – Parametergleichungen ..... 92  
 – Spline Modeler ..... 105  
 – Terrainmodellierung ..... 90  
 – Vertex Modeler ..... 112  
 CAS .. 173–175, 230, *siehe* Derive, Maple, Mathcad, Mathematica, MuPAD  
 – Vor- und Nachteile ..... 207–210  
 CATMULL-ROM-Splines ..... 103  
 CAUCHY, A.-L. .... 7  
 Caustics ..... 147  
 CAYLEY, A. .... 3, 7, 8  
 Charakter Animation ..... 157  
 CHOQUET, G. .... 18, 19  
 CIE (Commission Internationale de l’Eclairage) ..... 75  
 CIE-XYZ-Modell ..... 75  
 Cinema 4D ..... 144, 168, 170, 175  
 CLAIRAUT, A.-C. .... 3  
 CLIFFORD, W. K. .... 8  
 Clipping ..... 119



- clock** ..... *siehe* Animationen  
 CMY-Würfel ..... 77  
 CMYK-Modell ..... 77  
 Computeralgebrasystem ..... *siehe* CAS  
 Computergrafik-Pipeline 84, 116, 121, 160  
 Computerspiele ..... 50, 139  
 Computervideos ..... *siehe* Animationen  
**cone** ..... *siehe* Kegelstumpf  
 Container-Objekt ..... 95  
 Corel Designer ..... 70  
 Corel Draw ..... 70  
 Corel Photopaint ..... 68  
 CRAMER, G. .... 7  
 CSG (Constructive Solid Geometry) ... 89  
 CUNNINGHAM, S. .... 41  
 Curricula ..... *siehe* Rahmenpläne  
 curriculare Überlegungen .... 292–297, 327  
 Custom Tools ..... 58  
 Cyan ..... 77  
**cylinder** ..... *siehe* Zylinder
- D**  
 DA VINCI, L. .... 4  
 DANCKWERTS, R. .... 33  
 Datenstrukturen ..... 58, 194  
 Datentypen ..... 58, 194  
 DE-BOOR-Punkte ..... 100  
 DE-CASTELJAU-Algorithmus ..... 98  
**#declare** ..... 225  
 Derive ..... 173  
 DESCARTES, RENÉ ..... 2, 40  
 Descartes 3D ..... 176  
 Determinanten ..... 7  
 DGS (Dynamische Geometriesoftware) 71, 98, 176  
 DIEUDONNÉ, J. .... 18  
 Differenzmenge von Körpern ..... 89  
 Diffraction ..... *siehe* Beugung  
**diffuse** ..... 134  
 diffuse Reflexion ..... 131  
 Dimensionsformel ..... 5  
 Dimensionsverständnis ..... 37  
 Dini-Fläche ..... 155  
 Direct3D ..... 158, 161  
**direction** ..... *siehe* Kamera  
 Diskretisierung des Raumes ..... 112  
 Dispersion ..... 74, 144  
 Displacement Mapping ..... 135  
 DMV (Deutsche Mathematiker-Vereinigung) 20  
**double** ..... 194  
 dpi (dots per inch) ..... 67, 71  
 Drehung ..... 113, 257–259  
 Drehwinkel ..... 113, 115, 257–259  
 – Animation ..... 153, 259–260  
 DreiDGeo ..... 175, 205  
 Dreiecksnetz ..... 109–110, 136, 286–288  
 Durchschnittsmenge von Körpern ..... 89  
 DZUNG WONG, B. .... 47
- E**  
 Echtzeit-Animationen ..... 158–159  
 Echtzeit-Raytracing ..... 158  
 Eigenraum ..... 7  
 Eigenwert ..... 7  
 Einstieg in das Stoffgebiet Analytische Geometrie ..... 217–232  
 EISENSTEIN, F. G. M. .... 7  
 Elektrodynamik ..... 5  
 Ellipse ..... 262–264  
 Ellipsenbahnen ..... 252  
 Ellipsoid ..... 267  
 ELSCHENBROICH, H.-J. .... 182  
 ENDL, K. .... 181  
 Entfernung verdeckter Objekte ..... 121  
 Entwicklungsrichtungen für das Stoffgebiet Anal. Geom. .... 61, 292–297  
 – Grundkurse ..... 293  
 – Leistungskurse ..... 296  
 Environment-Map ..... 130  
 Eovia ..... *siehe* Carrara  
 EPA (Einheitliche Prüfungsanforderungen für das Abitur). 32, 34, 36, 57, 328  
 Erlanger Programm ..... 8  
 EUKLID ..... 58  
 Euklidischer Raum ..... 6

- EULER, L. . . . . 3, 7  
 Eulersche Winkel . . . . . 115, 195  
 experimentelles Arbeiten 56, 233, 249, 266,  
 274, 281, 283, 291  
 Extrusion . . . . . 104–105
- F**  
 Faltungsmatrizen . . . . . 68  
 Faltungsoperatoren . . . . . 68  
 Farbadddition . . . . . 76–77  
 Farbgamut . . . . . *siehe* Farbraum  
 Farbinterpolation . . . . . 136–138  
 Farbkreis . . . . . 79  
 Farbmodelle . . . . . 74–79  
 – CIE-XYZ-Modell . . . . . 75  
 – CMYK-Modell . . . . . 77  
 – HSB, HSV- und HLS-Modell . . . . . 78  
 – LAB-Modell . . . . . 79  
 – RGB-Modell . . . . . 76  
 – YIQ-Modell . . . . . 79  
 Farbpypamide . . . . . 79  
 Farbraum . . . . . 75  
 Farbsättigung . . . . . 78  
 Farbsechseck . . . . . 79  
 Farbsubtraktion . . . . . 77–78  
 Farbton . . . . . 78  
 Farbtripel . . . . . 76  
 Farbvektoren . . . . . 76–78, 195  
 Farbwahrnehmung . . . . . 74  
 Farbwiedergabe elektronischer Bildwieder-  
 gabegeräte . . . . . 75  
 Farbwürfel . . . . . 77  
 FERMAT, PIERRE DE . . . . . 2  
 Figurenarmut . . . . . *siehe* Formenarmut  
 FISCHER, R. . . . . 192  
 FLACHSMEYER, J. . . . . 46  
 Flächen  
 – Behandlung im Unterricht . 46, 265–274  
 – Bézierflächen . . . . . 106  
 – Dini-Fläche . . . . . 155  
 – Ellipsoid . . . . . 267  
 – Freiformflächen . . . . . 106–107  
 – Funktionsgraphen zweier Variablen . 46,  
 90, 269–270  
 – Hyperboloide . . . . . 267  
 – implizite Gleichungen . . . . . 91  
 – konische Spiralfäche . . . . . 92, 272–273  
 – kubische Flächen . . . . . 47  
 – Paraboloid . . . . . 267  
 – Parameterdarstellungen . . . . 91, 270–273  
 – Regelflächen . . . . . 268, 274  
 – Rotationsflächen . 91, 104, 268, 271, 272,  
 296  
 – Sattelfläche . . . . . 47, 267  
 – zweiter Ordnung . . . . . 91, 265–268  
 Flächendarstellung  
 – mathematischer Hintergrund . . . . 93–97  
 Flat Shading . . . . . 136  
 FLETCHER, T. J. . . . . 19  
 float (Fließkommazahl) . . . . . 194  
 Formenarmut des Unterrichts in analyti-  
 scher Geometrie . . . . . 18,  
 45–48  
 Fouriertransformationen . . . . . 69  
 FPS (Frames Per Second) . . . . . 151  
 fragend-entwickelnder Unterricht . . . . . 54  
 Frame . . . . . 151  
 FRANK, B. . . . . 2, 18  
 Freiformdeformation . . . . . 157  
 Freiformflächen . . . . . 106–107  
 Freiformkurven . . . . . 70, 98–103  
 Frequenzbereich . . . . . 69  
 FREUDENTHAL, H. . . . . 16, 19, 45  
 FÜHRER, L. . . . . 45, 48, 51, 191  
 fundamentale Ideen . . . . . 33, 36–39  
 Funktionalanalysis . . . . . 6  
 funktionaler Zusammenhang  
 – bei Parameterdarstellungen . . . . 38, 200  
 funktionales Denkens . . . . . 13, 14, 23, 252  
 Funktionenräume . . . . . 6  
 Funktionsgraphen zweier Variablen . . . . 90
- G**  
 GALILEI, G. . . . . 4  
 GAUSS, C. F. . . . . 7

- GAUSS-Algorithmus ..... 7, 60, 247–248  
 GDM (Gesellschaft für Didaktik der Mathematik).....177, 183  
 Gegenstandsaspekt .....58, 63, 189  
 Geometer's Sketchpad ..... 98, 177  
 Geometrisierung der Algebra.....9  
 Geosphäre .....88  
 Geschichte  
 – der analytischen Geometrie .....2–11  
 – – didaktische Schlussfolgerungen ..... 10  
 – des MU in anal. Geometrie ..... 12–24  
 geschlechtsspezifische Aspekte ..... 326  
 Gesellschaft Deutscher Naturforscher und Ärzte ..... 13  
 GIEDING, M. .... 45, 57, 180–181  
 Glättung .....136–138  
 Glanzpunkte ..... 132  
 Gleichungen geometr. Körper .... 221–224  
 globale Beleuchtung.....145–148  
 GNUPlot.....173  
 GOURAUD-Shading .....137  
 GPS (Global Positioning System).....49  
 Gradient .....95–97  
 Grafikkarte .....139  
 Grafikprozessor.....139  
 Grafiksoftware ..... 160–178  
 – Komponenten .....160–162  
 – mit grafischer Benutzeroberfl. . 166–170  
 – Schnittstellen ..... 160–162  
 – skriptgesteuerte .....162–165  
 Grafischer Taschenrechner .....315  
 GRASSMANN, H. G. .... 4–5  
 Graustufenbild ..... 67  
 Grunderfahrungen des MU ..... 32  
 Grundvorstellungen ..... 33, 37  
 Gruppen  
 – geometrischer Abbildungen ..... 7–8  
 GTR (Grafischer Taschenrechner) .... 315  
 GUDERIAN, D. ....184
- H**  
 Hüllkurven ..... 46  
 HABERSTROH, T. ....184
- HAMILTON, W. R. ....4–5  
 HAUBROCK, D. .... 49  
 Helligkeit .....78  
 Helligkeitsinterpolation ..... 136–138  
 – Behandlung im Unterricht ..... 286–288  
 HENN, H.-W. .... 33, 57, 59  
 Heuristik ..... 54  
 heuristische Strategien .....188, 290  
 heuristisches Arbeiten .....55, 56, 188  
 High-Level-API.....161, 162  
 Highlight-Vektor ..... 133  
 Highlights .....132  
 HILBERT, D. .... 6  
 Hintergrundfarbe .....81  
 Historische Entwicklung  
 – der analytischen Geometrie .....2–11  
 – – didaktische Schlussfolgerungen ..... 10  
 – des MU in anal. Geometrie ..... 12–24  
 Hochschulkurse zur Einführung in die Computergrafik.....186  
 HSB-Modell ..... 78  
 HSB-Pyramide ..... 79  
 Hue ..... *siehe* Farbton  
 HUSSMANN, S. ....54  
 HUYGENS, CHR. .... 4  
 Hyperbel ..... 262–264  
 Hyperboloide .....267
- I**  
 Ideen  
 – fundamentale .....33, 36–39  
 – universelle ..... 36–39  
 – zentrale .....36, 37  
 if ..... *siehe* Verzweigung  
 IGH (Internationale Gesamtschule Heidelberg) ..... 314  
 IMUK ..... 13, 18, 21  
 int (Integer) .....194  
 intentionale Probleme ..... 54  
 Interferenz .....74  
 Internationale Gesamtschule Heidelberg (IGH) 314  
 Interpolation .....103, 136–138

- inverse Kinematik ..... 157  
 Isosurface ..... 91, 97  
 iteratives Arbeiten ..... 57, 291
- J**  
 JavaView ..... 174, 175  
 Jittering ..... 150  
 JORDAN, C. .... 3, 7, 8  
 Jurassic Parc ..... 182
- K**  
 Kalkülorientierung ..... 33, 34, 54  
 Kamera ..... 81, 118, 213, 255, 257, 331  
 Kameraanimationen ..... 153  
 – Behandlung im Unterricht ..... 250–251, 253  
 Kamerakoordinatensystem ..... 116  
 Kantenglättung ..... 73, 136–138, 149  
 – Behandlung im Unterricht ..... 286–288  
 Kaustiken ..... *siehe* Caustics  
 Kegelgleichung ..... 223  
 Kegelschnitte ..... 15, 45, 261–264  
 Kegelstumpf ..... 87  
 Keyframes ..... 156  
 KLEIFELD, A. .... 47  
 KLEIN, F. .... 8, 13, 14, 24  
 KLINCSIK, M. .... 50  
 KMK-Bildungsstandards ..... 36  
 KMK-Empfehlungen (1968) ..... 19  
 Knoten (eines Sichtstrahlenbaums) ... 140  
 KÖBERLE, J. .... 314  
 Körpergeometrie (Software) ..... 177  
 Kommutativität ..... 113, 259  
 komplexe Zahlen ..... 4  
 Kongruenzabbildungen  
 – Matrizendarstellung ..... 114  
 konische Spirale ..... 152, 158, 251  
 konische Spiralfläche ..... 92  
 – Behandlung im Unterricht ..... 272–273  
 Konvertieren von Bildsequenzen in  
 Videodateien ..... 340  
 Koordinaten  
 – homogene ..... 3, 86, 114  
 – kartesische ..... 2  
 – Polarkoordinaten ..... 3  
 Koordinatengeometrie ... 14, 217–232, 249  
 – Aspekte ..... 10  
 Koordinatenmethode ..... 2–4, 40  
 Koordinatensystem ..... 85–86  
 – Ansichtskoordinatensystem ..... 116  
 – globales ..... 116  
 – Kamerakoordinatensystem ..... 116  
 – kartesisches ..... 85  
 – Kugelkoordinatensystem ..... 125  
 – linkshändiges ..... 85, 335, 343  
 – lokales ..... 116  
 – negativ orientiertes ..... 85  
 – Objektkoordinatensystem ..... 116  
 – rechtshändiges ..... 85, 335, 343  
 – Szenenkoordinatensystem ..... 116  
 – Weltkoordinatensystem ..... 116  
 Koordinatentransformation .. 83, 116–117,  
 343  
 – ungleichsinnige ..... 85  
 Koordinatisieren ..... 33, 36, 42  
 Kräfteparallelogramm ..... 4  
 Kreativität ..... 56  
 Kreis  
 – Gleichung ..... 221–222, 224  
 – Parameterdarstellung ..... 249  
 KROLL, W. .... 46  
 KRUMPE, N. .... 184  
 Kugel ..... 87  
 – Polygonnetz ..... 88  
 Kugelgleichung  
 – Anwendung für Positionierungen ... 225  
 – Herleitung ..... 222  
 Kugelkörper ..... 87  
 Kugelkoordinatensystem ..... 125  
 Kugeloberfläche ..... 87  
 Kultusministerkonferenz ..... *siehe* KMK  
 Kurven  
 – Archimedische Spirale ..... 152, 251  
 – Behandlung im Unterricht ..... 249–254  
 – Erzeugung durch Algorithmen ..... 10  
 – Hüllkurven ..... 46  
 – Kegelschnitte ..... 45, 261–264

- konische Spirale ..... 152, 158, 251
- Raumkurven ..... 46, 249–254
- Schraubenlinie ..... 46, 251
- zweiten Grades ..... 3

**L**

- LA HIRE, P. .... 3
- LACROIX, S. F. .... 4, 12
- Längenverhältnistreue ..... 125
- Lagebeziehungen
  - Ebenen im Raum ..... 245
  - Geraden im Raum ..... 237
  - Geraden und Ebenen ..... 242
- LAGRANGE, J. L. .... 3, 7
- LAMBERTSches Beleuchtungsmodell... 134
- LAMBERTSches Kosinusetz ..... 132
- Landschaftsmodellierung ..... 90
- LAPLACE, P. S. .... 7
- lathe ..... *siehe* Extrusion
- LEHMANN, E. .... 58, 60, 182, 184
- LEIBNIZ, G. W. .... 3, 4
- Leitideen ..... 36
- LENNÉ, H. .... 54
- Leuchtdichte ..... 131
- LEUDERS, T. .... 185, 289
- Lichtbeugung ..... *siehe* Beugung
- Lichtbrechung ..... *siehe* Brechung
- Lichteinfallvektor ..... 132
- Lichtquelle ..... 81, 128
- Lichtwellen ..... 74
- light\_source ..... *siehe* Lichtquelle
- Lightwave 3D ..... 168
- Lineale Ausdehnungslehre ..... 5
- Lineare Algebra
  - im Mathematikunterricht ..... 21–23
- Lineare Gleichungssysteme ..... 7
- Linke-Hand-Regel ..... 85
- Lissajous-Figuren ..... 250, 320
- Liste ..... 194
- location ..... *siehe* Kamera
- LOHMÜLLER, F. .... 207, 211
- lokale Beleuchtungsmodelle ..... 131–135
  - Behandlung im Unterricht ..... 283–286

- long (Long Integer) ..... 194
- look\_at ..... *siehe* Kamera
- Low-Level-API ..... 161
- LOWTHER, J. L. .... 186

**M**

- MAASS, K. .... 49, 201
- MACH-Band-Effekt ..... 137
- Macromedia Freehand ..... 70
- MAEDER, R. E. .... 109
- Magenta ..... 77
- MAJEWSKI, M. .... 184
- Makros ..... 58, 342–347
- MALLE, G. .... 192
- Mannigfaltigkeit ..... 8
- Maple ..... 173–175
- map\_type ..... *siehe* Texturprojektion
- Mathcad ..... 173
- Mathematica... 68, 92, 109, 138, 173–175
- MathGL3D ..... 174
- Matrizen ..... 7, 60, *siehe*
  - Transformationsmatrizen
- in Computeralgebrasystemen ..... 195
- in POV-Ray ..... 195
- in Programmiersprachen ..... 194
- Matrizendarstellung geometrischer
  - Transformationen ... 114–118, 255
- max\_gradient ..... *siehe* Gradient
- Maxon Cinema 4D ..... 144, 168, 170, 175
- max\_trace\_level ..... *siehe*
  - Strahlverfolgungstiefe
- Medienkompetenz ..... 189
- MEINERS, J.-C. .... 182
- Mental Ray ..... 144, 148, 165
- Meraner Reformen ..... 12–15, 23
- Merkmalsextraktion ..... 69
- mesh, mesh2 ..... *siehe* Dreiecksnetz
- metallic ..... 135
- MEYER, J. .... 46, 47
- Mikrofacetten ..... 135
- Mittelwertsatz ..... 96
- Modellbildungskreislauf ..... 33, 49
- Modellierung ..... 49, 56, 188

- Bewegungsbahnen ..... 249
- Doppelbedeutung ..... 50
- geometrische ..... 50, 217–232, 274
- mathematische ..... 49
- Modifikation
  - nicht-destruktive ..... 169
- modulares Arbeiten ..... 57, 188, 291
- MÖBIUS, A. F. .... 3, 4, 8
- Moiré-Effekt ..... 149
- MONGE, G. .... 4
- Moray ..... 170
- MÜLLER, J. .... 183–184
- MuPAD ..... 173–175
  - Animationen ..... 250
  - B-Splinekurven ..... 101
  - Bézierflächenstück ..... 107
  - Flächendarstellung
    - implizite Gleichungen ..... 91
    - Parametergleichungen ..... 92
  - geometrische Transformationen ..... 255
  - interpolierende Splinekurven ..... 103
  - Painter-Algorithmus ..... 122
  - Prozeduren ..... 100, 103, 107, 111, 175
  - Rotationsflächen ..... 105
  - Schneemann ..... 230
  - Szenenbeschreibungssprache ..... 163
  - Transformationsmatrizen ..... 256
  - Unterteilungs-Algorithmus ..... 111
  - Vor- und Nachteile ..... 207–210
- N**
  - $n$ -Tupel reeller Zahlen ..... 192
  - $n$ -Tupel-Modell ..... 195, 198–200
  - Nachbarschaftsoperationen ..... 68
  - Navigationssysteme ..... 49
  - Negativbilder ..... 68
  - Netzhaut ..... 74
  - Neue Mathematik ..... 18–21, 24
  - NEWTON, I. .... 3
  - nicht-destruktive Modifikation ..... 169
  - NICOL, M. .... 45
  - Normaleninterpolation ..... 136–138
    - Behandlung im Unterricht ..... 286–288
  - Normalenvektor ... 110, 132–134, 136–138, 275–288
    - Aufgaben ..... 279–280, 287–288
  - no\_shadow ..... 142
  - NURBS (Non-Uniform Rational B-Splines) ..... 102
- O**
  - Oberflächenstrukturen ..... 131–135
  - Objektkoordinatensystem ..... 83, 116
  - Objektliste ..... 119
  - objektorientierte Sichtweisen ..... 290
  - Objektraum-Algorithmen ..... 122
  - Objektstudien ..... 48, 60
  - Oktaeder ..... 286–288
  - OLDENBURG, R. .... 185, 285, 289
  - OpenGL ..... 158, 161
  - operative Begriffsbildung ..... 52
  - Orientierung ..... 85
  - Orientierungsfähigkeit ..... 228
  - orthographic.... *siehe* Parallelprojektion
  - Ortsvektoren ..... 234–235
  - OWEN, G. S. .... 186
- P**
  - Painter-Algorithmus ..... 122
  - PAPY, G. .... 18
  - Parabel ..... 262–264
  - Paraboloide ..... 267
  - Parallelprojektion ..... 118, 256–257
  - Parameter
    - Interpretation als Zeit ..... 201
  - Parameterdarstellungen ..... 3, 200–202
    - Animationen ..... 151–153, 249–254
    - dynamische Sicht ..... 201
    - Ebenen ..... 242–244
    - Flächen ..... 91, 270–273
    - funktionale Aspekte ..... 200
    - Geraden ..... 237–242
    - Raumkurven ..... 151–153, 249–254
    - Zylinder ..... 270
      - Variationen ..... 270–271
  - partielle Ableitungen ..... 53

- Partikelsystem ..... 157  
 PEANO, G. .... 5  
 Pfadanimationen ..... 156  
 Pfeilklassen ..... 17, 192, 198–199, 236  
**phong** ..... 134  
 PHONGsches Beleuchtungsmodell ..... 134  
 PHONG-Shading ..... 138  
**phong\_size** ..... *siehe* Rauigkeitsindex  
 Photonen ..... 74, 148  
 Photon-Tracing, Photon-Maps ..... 148  
 Photopaint ..... *siehe* Corel Photopaint  
 Photoshop ..... *siehe* Adobe Photoshop  
 Physiksimulationen .... 152, 157, 159, 252  
 PICKERT, G. .... 18  
 picture element ..... 67  
 Pixar RenderMan ..... 165  
 Pixel ..... 67  
 Pixelgrafik ..... 67–69  
 Pixelkoordinaten ..... 83, 123  
 pjBmp2Avi ..... 340  
 Plotter ..... 71  
 PLÜCKER, J. .... 3  
 Polarisisation ..... 74  
 Polarkoordinaten ..... 3  
 Polyeder ..... 87, 108  
 Polygonnetz ..... 87, 88, 108–112, 286–288  
 Polygonzug ..... 111  
 Polyray ..... 165  
 PoseRay ..... 175  
 Positionierungsaufgaben ..... 224  
 Post Processing ..... 186  
 POV-Ray ..... 164  
 – Animationen ..... 339  
 – Anleitungen ..... 211–213, 329–340  
 – Beispielszene ..... 80  
 – Berechnungen ..... 225  
 – Bibliotheken ..... 329, 341, 343  
 – Differenzmenge ..... 333  
 – Ebenen ..... 335  
 – Flächendarstellung ..... 337  
 – – implizite Gleichungen ..... 91, 337  
 – – mathematische Grundlagen ..... 94–97  
 – – Parametergleichungen ..... 92, 338  
 – Geraden ..... 335  
 – Include-Dateien ..... 329, 341, 343  
 – Kamera ..... 81, 331  
 – Koordinatensystem ..... 335  
 – Lichtquellen ..... 81  
 – Makros ..... 329, 341–347  
 – Prismen ..... 333  
 – Probleme mit der Syntax ..... 218, 325  
 – Punkte ..... 335  
 – Pyramiden ..... 333  
 – Rotationsflächen ..... 105  
 – Rotationskörper ..... 334  
 – Schnittmenge ..... 333  
 – Strecken ..... 335  
 – Texturen ..... 81, 131–135, 331, 336  
 – Transformationen ..... 332  
 – Transformationsmatrizen ..... 114  
 – Vektoren ..... 335  
 – Vereinigungsmenge ..... 332  
 – Videos ..... 339  
 – Vor- und Nachteile ..... 207–210  
 – Vorlagen ..... 211–213, 329–340  
**prism** ..... *siehe* Prisma  
 Prisma ..... 88, 104  
 PROFKE, L. .... 23, 24, 45, 192  
 Projektionen ..... 118–119, 256–257  
 – stereografische ..... 289  
 Projektionsebene ..... 118  
 Projektionszentrum ..... 118  
 Projektunterricht ..... 59, 296, 318–321  
 Prozeduren .... 58, 100, 103, 107, 111, 175  
 Pseudoanwendungsaufgaben ..... 49  
 Punktmengenauffassung geometrischer  
     Objekte 38, 200, 237, 242, 250–253  
 Pythagoras (Satz des) ..... 221, 227, 277  
 Python ..... 158, 163, 206  
**Q**  
 Quader ..... 87  
 quadratische Platkarte ..... 126  
 quadric ..... 143, 266  
 Quantenoptik ..... 74  
 Quaternionen ..... 5

- Querschnittskurven ..... 105  
 Querverbindungen .... *siehe* Vernetzungen
- R**
- Radiosity ..... 145–147  
 räumliches Strukturieren .. 33, 36, 42, 217  
 Rahmenpläne für das Stoffgebiet Analytische Geometrie ..... 25–31  
 – Baden-Württemberg ..... 30  
 – Berlin (ab 2004) ..... 29  
 – Berlin (Grundkurse) ..... 26  
 – Berlin (Leistungskurse) ..... 29  
 – Hamburg ..... 31  
 – Vorschläge für Änderungen ..... 327  
 Rastergrafik ..... 67–69  
 Rasterkonversion ..... 72  
 Rasterung ..... 71–73, 149  
 Rauigkeitsindex ..... 133  
 Raumkurven ..... *siehe* Kurven  
 Raumvorstellung ..... 13, 24, 40, 41, 187  
 – koordinatenbezogene ..... 42, 187, 217  
 – Teilfähigkeiten der Raumvorst. .... 42  
 Raytracing ..... 140–145  
 – Echtzeit-Raytracing ..... 158  
 Raytracing-Software  
 – mit grafischer Benutzeroberfl. .. 166–170  
 – skriptgesteuerte ..... 164–165  
**real** ..... 194  
 Rechte-Hand-Regel ..... 85  
 Reduktion  
 – zu betrachtender Dimensionen ..... 220  
**reflection** ..... 129, 134  
 Reflection-Map ..... 130  
 Reflexion ..... 74, 128–129  
 – diffuse ..... 131  
 Reflexionsgesetz ..... 128, 275–276  
 – Aufgaben ..... 281–283  
 Refraction-Map ..... 130  
 Refraktion ..... *siehe* Brechung  
 Regelflächen ..... 268, 274  
 Regenbogen ..... 144  
 REICHEL, H.-C. .... 192, 277  
 Rekonstruktion von 3D-Koordinaten aus Bildern ..... 289  
 Rekursionstiefe  
 – der Strahlverfolgung ..... 141–142  
 rekursives Arbeiten ..... 57, 291  
 Renderfarm ..... 144  
 Rendering ..... 84  
 Retina ..... *siehe* Netzhaut  
 RGB-Modell ..... 76  
 RGB-Vektoren ..... 76, 195  
 RGB-Würfel ..... 77  
 Richtungsvektoren ..... 234–235  
 RIEPER, F. .... 318  
**right** ..... *siehe* Kamera  
**rotate** ..... *siehe* Drehwinkel  
 Rotationsflächen ... 91, 104, 268, 271, 272, 296  
 Rotationskörper ..... 104, 298  
 ROTH, N. .... 47  
**roughness** ..... 135  
 Routineaufgaben ..... 34, 58, 233
- S**
- Sättigung ..... 78  
 Sattelfläche ..... 267  
 Saturation ..... *siehe* Sättigung  
 Satz des Pythagoras ..... 221, 227, 277  
**scale** ..... *siehe* Streckungsfaktor  
 Scan Conversion ..... 72, 123  
 Scanline Rendering ..... 139  
 Schärfen von Bildern ..... 68  
 Schatten ..... 142, 144  
 Schattierung ..... *siehe* Rendering  
 Scheitelpunktmodellierung .... 88, 108–112  
 Scherung ..... 115  
 Schleife ..... 152, 159, 237, 250, 291  
 Schlüsselszenen ..... *siehe* Keyframes  
 SCHMIDT, G. .... 34, 45, 52, 57  
 Schneemann ..... 217–230  
 Schnittberechnungen ..... 37  
 – mithilfe von CAS ..... 60  
 Schnittbetrachtungen ..... 37, 188, 220, 261–265, 267, 269, 270



- Schnittgeraden ..... 245  
 Schnittkurven... 53, 261–265, 267, 269, 270  
 Schnittpunkte  
 – von Geraden mit Ebenen ..... 242  
 – zweier Geraden ..... 240  
 schräger Wurf ..... 152, 159, 252  
 Schraubenlinie ..... 46, 251  
 SCHREIBER, A. .... 52  
 SCHREIBER, P. .... 10  
 SCHROTH, P. .... 46  
 Schülerkonzepte  
 – hinsichtlich des Vektorbegriffs ..... 193  
 – zu Parametergleichungen ..... 200  
 SCHULZ, CH. .... 181  
 SCHUPP, H. .... 35, 45, 50, 54, 62  
 SDL ... *siehe* Szenenbeschreibungssprache  
 Shadow-Maps ..... 144  
 shininess ..... 135  
 Sichtbarkeitsüberprüfung ..... 84, 121  
 Sichtfenster ..... 118  
 Sichtpyramide ..... 119  
 Sichtquader ..... 120  
 Sichtstrahlenbaum ..... 140  
 Sichtvolumen ..... 119  
 Signalverarbeitung ..... 69  
 Skalarprodukt ..... 132–134, 275–288  
 – Aufgaben ..... 278  
 – Einführung ..... 277–279  
 – visuell unterstützter Zugang ..... 278  
 Skalierung ..... 113, 257–259  
 – Matrizendarstellung ..... 115  
 Sketchpad ... *siehe* Geometer's Sketchpad  
 smooth\_triangle ..... *siehe*  
     Normaleninterpolation  
 Softimage XSI ..... 168  
 Software  
 – Auswahl für den Unterricht .... 203–210  
 Solarzelle ..... 285  
 Solid Primitives ..... 87  
 sor ..... *siehe* Extrusion  
 spekulare Highlights ..... 132  
 Sphärische Trigonometrie ..... 15  
 sphärische Projektion ..... 125  
 sphere ..... *siehe* Kugel  
 sphere\_sweep ..... *siehe* Extrusion  
 Spirale  
 – Archimedische ..... 152, 251  
 – konische ..... 152, 158, 251  
 Spiralprinzip ..... 51  
 Splinekurven ..... 100–103, 289  
 – Pfadanimationen ..... 156  
 Spurpunkte ..... 241  
 Standardaufgaben... 24, 34, 233, 239–241,  
     243–245, 278, 280  
 Standardinhalte des Stoffgebietes Analyti-  
     sche Geometrie ..... 233–248  
 STENDER, R. .... 16  
 STEVIN, S. .... 4  
 Storyboard ..... 156  
 Strahlenoptik ..... 74  
 Strahlverfolgung ..... *siehe* Raytracing  
 Strahlverfolgungstiefe ..... 141–142  
 Streckung ..... 113, 257–259, 268  
 – Animation ..... 153  
 – Matrizendarstellung ..... 115  
 Streckungsfaktor ..... 113  
 Streulicht ..... 131  
 string ..... 194  
 Subdivision Surface ..... 111  
 Supersampling ..... 149  
 SYLVESTER, J. J. .... 7  
 Synapsen ..... 74  
 Szene ..... 82  
 Szenenbeschreibung ..... 80  
 Szenenbeschreibungssprache ..... 162–165  
 Szenenkoordinatensystem ..... 116  
**T**  
 Tangentialebenen ..... 53  
 Teekanne ..... *siehe* Utah Teapot  
 Tension ..... 103  
 Tensorproduktflächen ..... 106  
 Terrainmodellierung ..... 90  
 Tessellation ..... 108  
 texture ..... *siehe* Texturen  
 Texture-Mapping ..... 124–127

- Texturen ..... 81, 130–135, 331, 336  
 Texturprojektion ..... 124–127  
 THURSTONE, L. L. .... 41  
 Tiefeninformationen ..... 121  
 TIETZE, U.-P. 15, 19, 20, 35, 36, 39, 41, 45, 46, 48, 55, 191, 193  
 TÖRNER, G. .... 22, 192  
 Tomographie ..... 112  
 Top-Down-Ansätze der Einführung in die Computergrafik ..... 186  
 Torus ..... 88  
 Transformationen ..... 113–121, 255–260  
 – zeitabhängige ..... 153–154, 259–260  
 Transformationsmatrizen ... 114–118, 256, 343  
 – homogene ..... 114–118  
**translate** ..... *siehe* Translationsvektor  
 Translationsvektor ..... 113, 114, 257–259  
 – Animation ..... 153, 259–260  
 Transparenz ..... 129–130  
 Treppeneffekt ..... *siehe* Aliasing  
**triangle** ..... *siehe* Dreiecksnetz  
 Triangulation ..... 109  
 Trickfilme ..... *siehe* Animationen  
 Tristimulustheorie ..... 74  
 TurboCad ..... 171–172  
 Turboplot ..... 173  
 Tweening ..... 156
- U**  
 Umgebungslicht ..... 131  
 universelle Ideen ..... 36–39  
 Unterrichtsformen ..... 54  
 Unterrichtskultur ..... 54–55, 326  
 Unterrichtsprojekte ..... 59, 318–321  
 Unterrichtsszenarien  
 – bei der Einbeziehung von Elementen der Computergrafik in den MU ..... 190  
 Unterrichtsversuch  
 – Grundkurs ma-3, Berlin ..... 300–313  
 – Grundkurs ma-3, Cottbus ..... 318–321  
 – Klasse 12, IGH ..... 314–318  
 Unterteilungs-Algorithmus ..... 111
- up** ..... *siehe* Kamera  
 Utah Teapot ..... 107, 126  
*u-v*-Mapping ..... 126
- V**  
 VANDERMONDE, A. T. .... 7  
 Variablenauffassung in der Informatik ..... 225, 290  
 Variablendeklarationen ..... 225, 229  
 Vectory ..... 175, 205  
 Vektoranalysis ..... 5  
 Vektorbegriff ..... 191–200  
 – in der Informatik ..... 194  
 Vektoren  
 – arithmetisches Modell (*n*-Tupel) ... 192, 195, 198–200  
 – geometrische Modelle ..... 191  
 – in Computeralgebrasystemen ..... 195  
 – in POV-Ray ..... 195  
 – in Programmiersprachen ..... 194  
 – Ortsvektoren ..... 234–235  
 – Pfeilklassenmodell ... 192, 198–199, 236  
 – Richtungsvektoren ..... 234–235  
 – Translationsmodell ..... 191  
 – Veranschaulichung durch Pfeile ..... 234  
 – Verbindungsvektoren ..... 234–235  
 Vektorgrafik ..... 70–73  
 Vektorisierung von Bildern ..... 69  
 Vektorräume ..... 191, 198  
 Vektorraumaxiome ..... 5  
 – im Mathematikunterricht ..... 18–21  
 Vektorrechnung ..... 4–5  
 – Einbeziehung in den MU ..... 16–17  
 Veranschaulichung ..... 56  
 – linearer Gleichungssysteme ..... 245–248  
 – nichtgeometrischer Objekte ..... 40  
 – Standardinhalte des Stoffgebietes Analytische Geometrie ..... 233–248  
 Verbindungsvektoren ..... 234–235  
 Vereinigungsmenge von Körpern ..... 89  
 Vernetzungen ..... 33, 51–53, 61, 188  
 – Analysis – analytische Geometrie ... 52, 254, 298

- horizontale ..... 52
- Mathematik-Informatik ..... 290–291
- Mathematik-Physik ..... 252
- vertikale ..... 51
- Verschiebungsvektor .... 113, 114, 257–259
- Animation ..... 153, 259–260
- Vertex Modeling ..... 88, 108–112
- Verzweigung ..... 154
- Videos ..... *siehe* Animationen
- virtuelle Realität ..... 50, 162
- Visualisierungen ..... 56, 60
- linearer Gleichungssysteme ..... 245–248
- Standardinhalte des Stoffgebietes Analytische Geometrie ..... 233–248
- VOLKERT, K. .... 40
- VOLLRATH, H.-J. .... 201
- Volumeneinheit ..... *siehe* Voxel
- Vorstellungsvermögen, räumliches ... *siehe* Raumvorstellung
- Vorwärts-Raytracing ..... 148
- Voxel ..... 112
- VPython ..... 158, 163, 206
- VRML ..... 161, 162
- W**
- WALLIS, J. .... 3
- Wavefront Object ..... 175
- Wavelet-Transformation ..... 69
- Weichzeichnen von Bildern ..... 68, 69
- WEIGAND, H.-G. .... 13, 33, 46, 58, 59
- Wellen
  - elektromagnetische ..... 74
- Wellenlänge ..... 74
- Wellenoptik ..... 74
- WELLER, H. .... 182
- Weltkoordinatensystem ..... 116
- Werkzeugaspekt ..... 58, 63, 189
- WESSEL, C. .... 4
- WETH, Th. .... 56, 59
- WEYL, H. .... 6
- #while** ..... *siehe* Schleife
- WHITEHEAD, A. N. .... 36
- WINTER, H. .... 32, 40
- WINTER, M. .... 46
- WITTMANN, G. .. 2, 9, 19, 37, 48, 193, 200
- WOLFE, R. .... 186
- WUNDERLING, H. .... 46, 182
- Wurfsimulation ..... 152, 159, 252
- X**
- X3D ..... *siehe* VRML
- XIANG, Z. .... 186
- Y**
- YafRay ..... 165
- YIQ-Modell ..... 79
- Z**
- z-Buffer-Algorithmus ..... 123
- Zeilenabtastung . *siehe* Scanline Rendering
- ZEITLER, H. .... 21
- zentrale Ideen ..... 36, 37
- zentrale Mathematisierungsmuster ..... 36
- Zentralprojektion ..... 118, 140, 256–257
- Ziele
  - der Einbeziehung von Elementen der Computergrafik in den MU ..... 187–189
- Zwischenwertsatz ..... 96
- Zylinder ..... 87
  - hyperbolischer ..... 266
  - Parameterdarstellung ..... 270
  - – Variationen ..... 270–271
  - verallgemeinerter ..... 104
- Zylinderprojektion ..... 125

# Selbstständigkeitserklärung

Hiermit erkläre ich, die vorliegende Arbeit selbstständig ohne fremde Hilfe verfasst und nur die angegebene Literatur und Hilfsmittel verwendet zu haben.

Andreas Filler  
13. April 2006